



**Universitat Autònoma
de Barcelona**

Millores en l'aplicació ekon SCRUM

Memòria del projecte

d'Enginyeria Tècnica en

Informàtica de Gestió

realitzat per

Ignacio Pablo Pérez Caro

i dirigit per

Jordi Pons Aróztegui

Escola d'Enginyeria

Sabadell, juny de 2011

El sotasignant, **Jordi Pons Aróztegui**,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en **Ignacio Pablo Pérez Caro**,

I per a que consti firma la present.
Sabadell, **Juny de 2011**.

Signat: **Jordi Pons Aróztegui**

El sotasignant, **Ramon Torres Arús**,
de Unit4,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en **Ignacio Pablo Pérez Caro**,

I per a que consti firma la present.
Sabadell, **Juny de 2011**.

Signat: **Ramon Torres Arús**

FULL DE RESUM – PROJECTE FI DE CARRERA DE L'ESCOLA D'ENGINYERIA

Títol del projecte: <i>Millores en l'eina ekon SCRUM.</i>	
Autor: <i>Ignacio Pérez Caro</i>	Data: <i>Juny 2011</i>
Tutor: <i>Jordi Pons Aróztegui</i>	
Titulació: <i>Enginyeria tècnica en Informàtica de Gestió</i>	
Paraules clau (mínim 3) <ul style="list-style-type: none">·Català: scrum, gestió, equips de desenvolupament.·Castellà: scrum, gestión, equipos de desarrollo.·Anglès: scrum, management, development teams.	
Resum del projecte (extensió màxima 100 paraules) <p>·Català: <i>Unit4 va començar a implantar en els seus equips de desenvolupament a nivell estatal la metodologia scrum. Amb els temps, es va tenir la necessitat de informatitzar el procés d'scrum. És per això que l'any 2009 es va iniciar el desenvolupament d'un software per aquest objectiu. L'actual projecte doncs, parteix de la necessitat d'acabar de desenvolupar algunes parts essencials per al bon funcionament del software ekon SCRUM.</i></p> <p>·Castellà: <i>Unit4 empezó a implantar en sus equipos de desarrollo a nivel estatal la metodología scrum. Con el tiempo, se tuvo la necesidad de informatizar el proceso de scrum. Es por eso que en 2009 se inició el desarrollo de un software para este objetivo. El actual proyecto parte pues, de la necesidad de finalizar el desarrollo de partes fundamentales para el buen funcionamiento del software ekon SCRUM.</i></p> <p>·Anglès: <i>Unit4 started to implement their development teams with scrum methodology statewide. Over time, there was the need to computerize the scrum process. That's why in 2009 began developing a software for this purpose. The current project starts with the need to finalize the development of key parts for the proper functioning of ekon SCRUM software.</i></p>	

Taula de continguts

1. Introducció.....	1
1.1. Objectius.....	1
1.2. Motivacions.....	2
1.3. Estat de l'art.....	2
1.4. Unit4.....	3
1.5. Organització del memòria.....	4
2. Introducció a SCRUM.....	5
2.1. Què és SCRUM?.....	5
2.2. Quins avantatges suposa?.....	5
2.3. Definicions, acrònims i abreviacions.....	7
2.4. Com funciona SCRUM? La seva lògica.....	7
3. Estudi de viabilitat.....	13
3.1. Introducció.....	13
3.1.1. Objectius del projecte.....	13
3.1.2. Parts interessades.....	14
3.2. Estudi de la situació actual.....	14
3.2.1. Context.....	14
3.2.2. Descripció física.....	14
3.2.3. Usuaris i/o personal del sistema.....	15
3.2.4. Diagnòstic del sistema.....	15
3.3. Requisits.....	16
3.3.1. Requisits funcionals.....	16
3.3.2. Requisits no funcionals.....	19
3.3.3. Catalogació i priorització dels requisits.....	20
3.4. Alternatives a la solució.....	21
3.5 Conclusió.....	21
4. Pla de projecte.....	23
4.1. Introducció.....	23
4.2. WBS.....	23
4.2.1. Fases i activitats del projecte.....	23
4.2.2. Milestones.....	24
4.3. Recursos del projecte.....	24
4.4. Calendari del projecte.....	25

4.4.1. Dependències.....	25
4.4.2. Quadre de tasques del projecte.....	25
4.4.3. Calendari temporal.....	27
4.7. Pressupost.....	27
4.7.1. Estimació cost de personal.....	28
4.7.2. Estimació cost dels recursos.....	28
4.7.3. Estimació cost de les activitats.....	29
4.7.4. Resum i anàlisi cost benefici.....	30
4.6. Conclusions.....	30
5. Anàlisi i disseny.....	31
5.1. Introducció.....	31
5.2. Arquitectura de l'aplicació.....	31
5.3. Interfície d'usuari.....	32
5.4. Funcionalitats de les millores.....	34
5.5. Diagrama de la base de dades.....	44
6. Implementació.....	47
6.1. Introducció.....	47
6.2. Entorn de desenvolupament.....	47
6.3. Entorn d'execució.....	48
6.4. Desenvolupament en Java.....	49
7. Codificació i proves.....	51
7.1. Introducció.....	51
7.2. Estil de codificació.....	51
7.3. Proves unitàries.....	52
7.4. Test en departament I+D.....	56
8. Valoració del projecte final de carrera.....	57
8.1. Objectius aconseguits i no aconseguits.....	57
8.2. Possibles ampliacions i millores.....	58
8.3. Desviacions respecte la planificació.....	59
8.4. Valoració personal.....	60
Bibliografia.....	61
Contingut del CD.....	62

Índex de taules

Taula 3.1: Valoració dels objectius del projecte.....	13
Taula 3.2: Parts interessades.....	14
Taula 3.3: Priorització dels requisits funcionals.....	20
Taula 3.4: Priorització dels requisits no funcionals.....	21
Taula 4.1: Fases generals del projecte.....	23
Taula 4.2: Milestones o punts claus del projecte.....	24
Taula 4.3: Valoració dels diferents recursos del projecte.....	24
Taula 4.4: Planificació prevista.....	26
Taula 4.5: Resum del cost del personal.....	28
Taula 4.6: Resum del cost dels recursos.....	28
Taula 4.7: Resum del cost de les activitats.....	30

Índex de figures

Figura 2.1: Reunions i rols en SCRUM.....	9
Figura 4.1: Diagrama de Gantt del projecte.....	27
Figura 5.1: Dissenyadora de formularis.....	32
Figura 5.2: Vista d'un formulari Karat.....	33
Figura 5.3: Formulari de l'assistent d'sprints.....	36
Figura 5.4: Formulari de la importació de històries.....	37
Figura 5.5: Formulari de la duplicació de tasques.....	39
Figura 5.6: Formulari de la consulta d'hores per tècnic i dia.....	40
Figura 5.7: Formulari de creació de històries.....	41
Figura 5.8: Formulari del històric dels camps a guardar.....	41
Figura 5.9: Formulari de la visió global dels sprints.....	42
Figura 5.10: Formulari de la imputació d'hores a tasques.....	44
Figura 5.11: Diagrama de les taules usades.....	45
Figura 6.1: Mètodes a desenvolupar segons els events que genera karat-.....	50
Figura 6.2: Propietats de l'Objecte de negoci.....	50
Figura 8.1: Diagrama de Gantt de la planificació final.....	49

1. Introducció

1.1. Objectius

Aquest projecte ha estat realitzat en col·laboració amb l'empresa UNIT4, d'acord al conveni de cooperació entre la entitat i la Universitat Autònoma de Barcelona (UAB).

L'aplicació ekon SCRUM va sorgir de la necessitat de l'empresa d'informatitzar tota la metodologia i conjunt de bones pràctiques englobades en el que s'anomena metodologia SCRUM. La necessitat de l'empresa de coordinar informàticament i deixar constància de la gestió dels diferents equips de desenvolupament en el procés que es porta a terme en la metodologia SCRUM. Aquesta metodologia fa incorporar a l'equip de desenvolupament un aspecte molt més global del projecte, obtenint un control complet tant de la feina pròpia del desenvolupador com la dels altres companys d'equip. Això doncs, provoca una probabilitat d'èxit elevat en la conclusió del projecte en les dates acordades.

L'objectiu d'aquest projecte, tal i com indica el títol, és el de millorar tots els aspectes possibles de l'aplicació **ekon SCRUM**. Hi ha diversos aspectes de l'aplicació que no són còmodes per a l'usuari i tenen una lentitud excessiva sobretot alhora d'introduir dades i poder treballar amb elles.

ekon SCRUM és un programa intern de l'empresa que plasma tot el control de la metodologia scrum en un software per tal de tenir una sèrie de registres informàtics navegables i consultables en el temps.

Tenint en compte la naturalesa del projecte, que no és un altra que la de millorar una aplicació existent, estem doncs parlant d'un projecte obert. Això implica que els objectius del projecte poden ser molt diferents, i poden implicar el desenvolupament de noves funcionalitats de l'aplicació, o bé, la millora d'altres parts del software existent.

La fita primordial del projecte es que a la finalització del mateix l'aplicació **ekon SCRUM** sigui una aplicació usable, amigable, que millori notablement l'experiència de l'usuari i que tingui un funcionament ràpid i àgil

El projecte ha estat realitzat entre novembre de 2010 i juny de 2011, amb una duració total, de 560 hores, regulades pel conveni citat anteriorment.

Per la part tècnica, l'entorn de desenvolupament ha estat karat 8.0, plataforma tecnològica creada i desenvolupada per UNIT4 que permet crear, afegir, personalitzar productes o millorar-ne d'existents amb noves funcionalitats segons esdevingui la necessitat de fer-ho.

Absolutament tota la plataforma karat, funciona amb Java, per tant és necessari l'ús del mateix per desenvolupar i implementar millores sobre la citada plataforma.

1.2. Motivació

La principal motivació per realitzar aquest projecte és el fet de poder viure una experiència laboral en el món de la informàtica professional. Des de que vaig conèixer la possibilitat de realitzar un conveni de pràctiques entre la Universitat Autònoma de Barcelona i Unit4 els meus esforços es van centrar en poder acabar totes les assignatures de la carrera per poder dedicar-me completament a fer el projecte en aquesta empresa.

Crec que és molt important poder finalitzar els meus estudis d'aquesta manera, i alhora demostrar fora del món acadèmic les meves aptituds i coneixements.

1.3. Estat de l'art

En els meus estudis inicials de l'aplicació i en les primeres reunions amb el meu tutor del projecte, vaig poder comprovar de primera mà quins eren els punts febles del software.

L'aplicació ekon SCRUM va ser desenvolupada per quatre becaris de l'any 2009/2010. A la finalització del projecte és possible que quedessin inacabats certs aspectes de l'aplicació tot i que és una aplicació que té una integritat de dades perfecte, i quasi no hi ha cap aspecte a millorar en relació a les bases de dades. Les millores doncs han de ser aplicades i desenvolupades en noves funcionalitats que afegixin usabilitat a l'aplicació. Aquestes millores poden incloure assistents, escriptoris, automatitzacions en modificació de dades, etc.

La necessitat de poder manejar les dades amb rapidesa va portar a fer servir Microsoft Excel en detriment de ekon SCRUM per a explotar les dades que anteriorment havien estat entrades a l'aplicació. Aquest full Excel obtenia les dades de les bases de dades de l'aplicació, i es feien servir per al procés d'scrum en comptes de la pròpia aplicació dissenyada exclusivament per a això.

És per aquest motiu que moltes de les noves funcionalitats que seran desenvolupades són idees extretes de l'ús intensiu que els treballadors de la planta de I+D han anat experimentant amb aquests fulls de càlcul Excel.

1.4. Unit4

Unit4 és una multinacional d'origen holandès del sector de les tecnologies de la informació que ofereix diverses solucions *ERP*¹ per a la gestió total de les empreses i serveis associats. Les seves oficines centrals es troben a Slidrecht (Holanda).

Unit4 desenvolupa, implementa i ven diverses suites ERP, que poden estar formades per Finances, Logística, Comptabilitat, Recursos Humans, Gestió de Projectes, Nòmina, CRM, etc..

Fabrica i comercialitza software de gestió empresarial al que a més a més ofereix suport mundialment de forma àgil i flexible, amb l'objectiu d'ajudar a les organitzacions dinàmiques a gestionar les seves necessitats de negoci de manera efectiva.

Unit4 disposa de filials en 17 països europeus, així com en altres 7 repartits entre Amèrica, Àsia-Pacífic i Àfrica.

La firma d'un conveni de pràctiques en la seu de Barberà del Vallés entre Unit4 i la UAB permet que diferents alumnes realitzin el projecte a l'empresa. En el cas del projecte "Millores en l'eina ekon SCRUM" és realitzat al departament I+D.

¹ ERP: de l'anglès, Enterprise Resource Planning

1.5. Organització de la memòria

Primer capítol: Introducció, conté una visió general del projecte, així com dels motius personals que han dut a terme la realització d'aquest projecte.

Segon capítol: Introducció a SCRUM, en el qual s'explica els temes més importants per a poder entendre SCRUM de manera òptima.

Tercer capítol: Estudi de viabilitat, en el qual parlarem sobre els objectius plantejats, de la situació que volem millorar i de les diferents alternatives que trobem.

Quart capítol: Pla de projecte, en aquest capítol parlarem de la planificació del projecte, de l'avaluació de riscos, i del pressupost.

Cinquè capítol: Anàlisi i disseny, parlem del context funcional de les millores i les seves funcionalitats.

Sisè capítol: Implementació, del software, els seus entorns de desenvolupament, d'execució, i del desenvolupament en Java.

Setè capítol: Codificació i proves, en el qual parlarem de les proves i de l'estil de codificació.

Vuitè capítol: Valoració del projecte final de carrera, en aquest capítol parlarem dels beneficis que s'han obtingut al realitzar el projecte a Unit4, tant per la banda personal com empresarial.

2. Introducció a SCRUM

2.1. Què és SCRUM

SCRUM és un conjunt de bones pràctiques dins de l'àmbit de la programació, sovint denominat metodologia. Tenint en compte les dues premisses anteriors, podem dir que:

SCRUM és una metodologia per a la gestió i desenvolupament de software basada en un procés iteratiu i incremental utilitzat normalment en entorns productius basats en el desenvolupament àgil de software.

Una altra definició alternativa diria que SCRUM és un model àgil de desenvolupament de software.

Com a nota interessant cal remarcar que la paraula SCRUM prové de l'esport del rugby per definir la formació "espatlla amb espatlla" que permet moure la pilota cap endavant de manera conjunta.

Aquesta paraula va ésser usada per primera vegada per *Takeuchi y Nonaka*² en un famós article publicat en la "*Harvard Business Review*" en el que es descriuen els projectes amb més èxit de desenvolupament de productes al Japó.

2.2. Quins avantatges suposa?

Els principals beneficis que proporciona SCRUM són:

- Gestió regular de les expectatives del client:

El client estableix les seves expectatives indicant el valor que li aporta cada requisit del projecte i quant espera que estigui acabat. Podríem dir que el client fixa una puntuació inicial i unes dates de finalització per a cada requisit. A diferència dels altres mètodes de gestió de projectes en els quals el client només és part activa a l'inici i al final del desenvolupament del projecte, en SCRUM, els clients comproven de manera regular si es van complint les seves expectatives, i a més a més va proporcionant un *feedback*³ des de l'inici del projecte. Alhora el client pot prendre decisions a partir de resultats objectius i dirigeix els resultats dels projecte, iteració a iteració, cap a la finalització del mateix.

² Takeuchi and Nonaka: The New Product Development Game (Harvard Business Review, Jan-Feb 1986)

³ Feedback: de l'anglès **realimentació**. És el procés de compartir observacions i suggeriments entre dues parts per millorar el funcionament.

- Resultats anticipats (*Time to market*):

Un cop s'ha realitzat la puntuació dels requeriments per part del client i de l'equip del projecte, l'ordre de desenvolupament d'aquests requisits ve definit per la puntuació i la importància. Per tant a l'inici del projecte les parts més interessants pel client són desenvolupades, i es poden començar a testar i fer servir, abans de que arribi la data de finalització del mateix.

- Flexibilitat i adaptació:

De manera regular el client redirigeix el projecte en funció de les seves noves prioritats, dels canvis als mercats, dels requisits completats que li permeten entendre millor el producte, de la velocitat real de desenvolupament, etc... Al final de cada iteració el client pot aprofitar la part del producte completat fins a aquest moment per fer proves de concepte amb usuaris o consumidors i prendre decisions en funció del resultat obtingut.

- Gestió sistemàtica del Retorn de la Inversió (ROI):

De manera regular, el client maximitza el ROI del projecte. Quan el benefici pendent d'obtenir és menor que el cost de desenvolupament, el client pot finalitzar el projecte.

- Mitigació sistemàtica dels riscos del projecte:

Des de la primera iteració l'equip ha de gestionar els problemes que apareguin en una entrega del projecte. Com que es tenen presents aquests riscos, és possible iniciar la seva mitigació de manera anticipada. El *feedback* constant amb el client permet estalviar esforç i temps en errors tècnics.

- Productivitat i qualitat:

De manera regular l'equip va millorant i simplificant la seva forma de treballar.

Els membres de l'equip sincronitzen el seu treball diàriament i s'ajuden a resoldre els problemes que poden impedir aconseguir l'objectiu de la iteració. La comunicació i l'adaptació a les diferents necessitats entre els membres de l'equip són màximes, de manera que no siguin realitzades tasques innecessàries.

- Alineació entre client i equip:

Els resultats i esforços del projecte es mesuren en forma d'objectius i requisits entregats al negoci. Tots els participants en el projecte coneixen quin és l'objectiu a aconseguir. El producte s'enriqueix amb les aportacions de tots.

- Equip motivat:

Els desenvolupadors de l'equip estan més motivats quan poden usar la seva creativitat i intel·lecte per resoldre problemes i poden decidir la seva manera d'organitzar-se.

Cal recordar que en els equips SCRUM, no hi ha un cap visible dintre del desenvolupament, sinó que tots estan al mateix nivell.

2.3. Definicions, acrònims i abreviacions

A continuació es podrà trobar una sèrie de definicions, acrònims i abreviacions per tal de poder entendre amb més claredat posteriors explicacions de la metodologia SCRUM i altres seccions d'aquesta memòria.

- a) Sprint: iteració. Cada un dels cicles de desenvolupament, de duració determinada entre tres i quatre setmanes. A l'sprint són desenvolupades les històries o *product backlog*.
- b) Històries: un requisit del software a implementar.
- c) Product Baklog: conjunt de les històries que defineixen tot un software a desenvolupar.
- d) Tasques: les divisions d'una història. Les tasques són la feina indivisible a realitzar per un únic tècnic.

2.4. Com funciona SCRUM? La seva lògica

Abans de poder explicar la metodologia SCRUM, és necessari parlar dels diferents rols que existeixen.

En aquesta metodologia o model àgil, apareixen els següents rols: *Customer* (Client), *Product Owner* (Client propietari), *Scrum Master* (Facilitador) i *Team members* (Membres de l'equip). Aquests mots no són traduïts al castellà o català, sino que són usats normalment en anglès.

1. Customer:

El Customer o client, és la persona que encarrega el projecte. Està situat a un nivell superior, ja que és l'encarregat d'indicar a l'equip realitzador/desenvolupador del projecte, quins són els seus requisits i té potestat per canviar-los tot i que el projecte ja hagi començat. Per tant, a diferència d'altres mètodes de desenvolupament, el client forma part activa en tot el procés.

2. Product Owner:

El Product Owner (PO) és el representant de totes les persones interessades en el projecte (internes o externes a la organització, promotors i usuaris finals). Actua com interlocutor únic de les persones interessades davant de l'equip i té autoritat per prendre decisions. Dintre de l'equip de desenvolupament, el PO ha de ser una persona que conegui i domini el producte, el sector, el mercat i/o la competència. El Product Owner és l'encarregat de recollir els requisits, requeriments i desitjos del Client. Per tant, el PO ha de definir els objectius i dirigir els resultats del projecte, i maximitzar

el ROI (Retorn de la Inversió). Ha de ser el propietari de la planificació del projecte, i sempre ha d'estar disponible per respondre i resoldre dubtes a l'equip.

3. Scrum Master:

L'Scrum Master (SM), actua com a líder de l'equip. Les seves responsabilitats són molt àmplies. El SM ha de vigilar que els participants del projecte segueixin les regles i processos d'SCRUM.

A més a més, és la persona que facilita les reunions de manera que aquestes ajudin a aconseguir els objectius i que siguin profitoses. El SM verifica constantment que l'equip estigui compost per perfils adequats que permetin realitzar les missions encomanades i alhora ha d'ensenyar a l'equip a autogestionar-se. De fet, una de les màximes del SM és la de no donar respostes, sino formular preguntes amb l'objectiu de guiar l'equip cap a la bona solució. En definitiva, ha de treure tots els impediments que l'equip pot tenir en el seu camí per aconseguir l'objectiu de cada sprint. Aquests obstacles s'identifiquen de manera sistemàtica en diferents reunions. El SM és, per tant, el cap visible de l'equip SCRUM i d'aquesta manera protegeix i aïlla l'equip d'interrupcions externes durant l'execució de la iteració.

4. Scrum Team:

L'equip de desenvolupament actua com a una única unitat, tal i com s'extreu del mot SCRUM (*ref: 1.1.1. "Què és Scrum?" pp. 1*). L'equip desenvolupa els requisits de manera que estiguin preparats per ser entregats. D'aquí es pot extreure un dels beneficis principals citats anteriorment: A cada sprint, el que s'hagi desenvolupat i testejat, ha de ser entregable. L'equip és l'encarregat d'aproximar la complexitat de cada història en les reunions de planificació de l'sprint, i alhora s'ha de comprometre a completar tots i cada un dels sprints del projecte. Un cop s'hagi acabat cada iteració, s'ha de demostrar tot el que s'ha fet. La millora sistemàtica i diària del grup de desenvolupament es realitza gràcies a les reunions diàries i les del final de cada sprint. L'equip ha d'estar format per diferents perfils. Per tant, ha d'existir un analista, desenvolupadors varis, un testejador i un documentalista.

A continuació trobem un esquema del funcionament de les reunions que existeixen en SCRUM:



Figura 2.1. Reunions i rols en SCRUM

Les reunions que es poden observar a la *Figura 1* són les de *Sizing*, *Release Planning*, *Sprint pre Planning*, *Sprint Daily Scrums*, *Sprint Review* i *Sprint Retrospective*.

Fixem-nos que inicialment el Product Owner (PO) manté una reunió inicial amb el client, en la qual són recollits el punt de vista del client, els seus requeriments i paràmetres varis (temps, dates d'entrega, revisions, reunions varies, etc...).

Quan el PO ha analitzat el projecte, ha definit els diferents requisits del client en les anomenades històries i examina la viabilitat del projecte, aquest és acceptat i passa a formar part d'un equip de desenvolupament SCRUM.

La primera reunió s'anomena *Sizing*, i té com a objectiu votar la complexitat de les històries, en punts. Aquesta reunió inicial no té una durada determinada ja que pot durar dues, tres, quatre hores o inclús més, perquè cada història ha de ser votada per tot l'equip i només se li assigna la puntuació quan hi ha

convergència i unanimitat en aquesta. Això pot provocar que l'equip discuteixi durant un determinat temps la dificultat de cada història. D'aquesta primera reunió d'equip se n'extreu l'anomenat Product Backlog.

La següent reunió que es realitza s'anomena *Release Planning*. Aquesta reunió, com a molt, ha de durar vuit hores i durant la mateixa l'equip assigna les històries del Product Backlog als diferents sprints, en base al número de punts (velocitat) planificada per a cada sprint. A partir d'aquest moment, l'sprint es dona per començat.

La primera reunió que pròpiament es troba en l'àmbit de l'sprint es diu *Sprint pre Planning*, amb una duració màxima de dues hores. En aquesta reunió l'equip ha de confirmar el número de punts disponibles i decidir si hi ha algun detall de l'sprint a canviar.

Acte seguit a la finalització de l'*Sprint pre Planning*, comença la *Sprint Planning*, ja sense el Product Owner, en la qual s'analitzen totes les històries contingudes en l'sprint actual, i són dividides totes aquestes històries en les anomenades tasques. Quan es creen aquestes tasques, també s'ha de fer una estimació de les hores necessàries per tasca. Un cop s'han realitzat totes les reunions anteriors, l'sprint pot començar. Normalment, l'sprint té una duració d'entre 15 a 30 dies. Per exemple, a Unit4, els sprints tenen una durada de tres setmanes. Durant la durada de l'sprint, l'equip va convertint el Product Backlog en funcionalitat i cada dia, normalment a la mateixa hora, té lloc una petita reunió, en la qual, cada membre de l'equip ha de respondre les preguntes, "Què vaig fer ahir?", "Què faré avui?" i "Quins problemes tinc?". Aquestes reunions diàries són anomenades *Daily Scrum*.

Quant l'sprint ha estat finalitzat, l'equip ha de presentar totes les tasques de les històries desenvolupades al Product Owner i, si s'escau, al client. Aquesta reunió s'anomena *Sprint Review*. Com que en aquest instant l'sprint ha finalitzat, el client se'n pot endur una versió del producte desenvolupat, per poder fer-ne proves o, fins i tot, fer-lo servir si el projecte es troba en fase de finalització.

Per últim, la reunió anomenada *Sprint Retrospective* serveix per analitzar problemes, trobar àrees que s'hagin de millorar i acordar diferents accions.

Per tant, i a mode de conclusió, podem dir que els pilars que formen scrum, són les reunions, i el fet de que l'equip avança de manera uniforme i unitària segons les seves habilitats i "sense la pressió" d'un cap o responsable que supervisi diàriament la feina realitzada.

3. Estudi de viabilitat

3.1. Introducció

En aquest capítol s'analitzarà la viabilitat del projecte. A través dels diferents apartats d'aquest capítol quedarà especificada la seva viabilitat tècnica i operativa.

3.1.1. Objectius del projecte

Els objectius del projecte són els següents. Com es podrà veure a la taula 3.1, l'objectiu 8 és secundari i està sotmès a una anàlisi exhaustiu de la viabilitat i l'ús que pot tenir l'escriptori en qüestió.

Objectiu 1: Assistent de Sprints.

Objectiu 2: Importador d'històries des de Microsoft Excel.

Objectiu 3: Duplicació de Tasques.

Objectiu 4: Consulta de la imputació d'hores realitzades per tècnic/dia.

Objectiu 5: Control de modificacions de les històries.

Objectiu 6: Visió global Gantt dels sprints.

Objectiu 7: Automatització de l'estat de les tasques/històries segons hores imputades.

Objectiu 8: Escriptori del Product Owner.

A continuació trobem un quadre amb les valoracions per prioritat de cada un dels objectius:

	Crític	Prioritari	Secundari
Objectiu 1	X		
Objectiu 2		X	
Objectiu 3			X
Objectiu 4		X	
Objectiu 5		X	
Objectiu 6		X	
Objectiu 7	X		
Objectiu 8			X

Taula 3.1. Valoració dels objectius del projecte

3.1.2. Parts interessades

El projecte té definides unes parts interessades o *stakeholders*. Per una part tenim l'empresa Unit4, la qual explotarà l'eina a desenvolupar. També tenim definit el cap del projecte, que és l'encarregat de vetllar per la finalització del projecte que s'està desenvolupant fent servir l'eina ekon SCRUM. Per l'altre banda tenim els desenvolupadors, que tindran amb aquesta eina un control exhaustiu de la feina realitzada per ells i els seus companys i veure com el projecte va evolucionant. A més a més, la direcció de l'empresa té la necessitat de conèixer les hores realitzades pels treballadors. En aquest cas, treballadors del departament de I+D.

Perfil	Responsabilitat
Empresa Unit4	Explotació del software tant a la seu de Barberà del Vallés com a nivell mundial a les seves filials.
Caps de projecte	Gestió del projecte del qual es porta el control.
Desenvolupadors	Poder recollir la informació de les tasques a realitzar en cada moment i poder saber quin és l'avanç d'altres tasques relacionades
Direcció executiva de l'empresa	Saber en tot moment quines són les hores reals realitzades pels treballadors del departament I+D.

Taula 3.2. Parts interessades

3.2. Estudi de la situació actual

3.2.1. Context

Actualment l'empresa disposa de la base del software implementat. Concretament aquest software va formar part del Projecte Final de Carrera de quatre alumnes becaris de la Universitat Autònoma de Barcelona. Tot i que la integritat de les dades és molt forta i darrera hi ha una gran feina a nivell de base de dades, algunes parts del programa no són del tot operatives.

3.2.2. Descripció física

Com que l'aplicació està desenvolupada en Java, no hi ha requisits tècnics en quan a plataforma operativa, és a dir, es pot fer servir Microsoft Windows, Mac OS X, o qualsevol distribució GNU/Linux sempre i quan es disposi del client de Java Virtual Machine instal·lat al terminal.

Pel que fa al servidor a on es guarden les dades, la plataforma ekon ofereix suport per a molts dels sistemes de bases de dades, per poder adaptar-se a cada client. Aquests sistemes de bases de dades poden ser: MySQL, Oracle, Intersystems Cache, SQL Server. Si que es recomana la utilització de Microsoft Windows 2003 Server Edition com a sistema operatiu del servidor.

3.2.3. Usuaris i/o personal del sistema

Els usuaris del sistema seran els treballadors de l'empresa Unit4, a nivell internacional, ja que serà fet servir per tots els departaments de les empreses del grup, que treballin amb la metodologia SCRUM. Es preveu que el primer semestre de 2011 totes les empreses del grup treballin amb metodologia SCRUM.

A nivell genèric els usuaris de l'aplicació seran els mateixos, que els rols que figuren en la metodologia SCRUM:

- a) Product Owner.
- b) Scrum Master.
- c) Team Members.

3.2.4. Diagnòstic del sistema

Ara per ara, la metodologia SCRUM és molt dependent dels fulls de càlcul de Microsoft Excel. Per exemple, alhora de planificar els sprints, el Scrum Master realitza les assignacions d'històries de cada sprint en una fulla d'Excel que té connexió a les taules de l'aplicació ekon SCRUM.

També, per part de direcció, és difícil, portar un control exhaustiu de les hores treballades pels desenvolupadors, ja que s'haurien de desplaçar als diferents departaments per mirar els panells. És per aquest motiu pel qual es va desenvolupar i es vol millorar, una eina que informatitzi tot el procés de desenvolupament de cada grup, per tal de que quedi un registre extra en aquest procés. Per tant, es vol primar l'accessibilitat de les dades, la velocitat de la consulta i la interacció entre les parts.

3.3. Requisits

3.3.1. Requisits funcionals

Tot projecte que es vol desenvolupar té uns requisits funcionals. En el cas d'aquest projecte són:

Objectiu 1: Assistent d'sprints.

Es vol poder assistir el màxim possible la previsió dels sprints. És per això que s'ha de crear aquest assistent/simulador que permeti associar de manera ràpida i còmode diferents històries amb cada sprint. Aquest assistent, ha de permetre doncs, tant la inserció d'històries, com la eliminació d'històries ja assignades. En definitiva, aquest ball d'històries d'un sprint a l'altre és el que fa dotar a aquest formulari la funcionalitat de simulador o assistent de creació d'sprints.

Els requisits són doncs:

- 1.a) Definir relació un a molts entre un sprint i vàries històries.
- 1.b) Creació d'un formulari amb grid⁴ per a poder afegir històries.
- 1.c) Controls dinàmics en associació d'històries.
- 1.d) Automatització dels punts disponibles per a l'assignació d'històries a cada sprint.
- 1.e) Creació de la consulta d'històries disponibles per afegir a cada sprint.
- 1.f) Impressió del formulari en format "pdf". Capçalera amb dades de l'sprint i grid amb les històries.

Objectiu 2: Importador d'històries des de Microsoft Excel.

Com hem vist en el capítol de les reunions a la metodologia scrum (Cap. 2.1.4), l'equip de desenvolupament es reuneix a l'inici del projecte en una reunió anomenada Sizing. D'aquesta reunió se n'extreuen totes les històries a desenvolupar. És habitual que l'Scrum Master generi un fitxer Excel per anar guardant totes aquestes històries. En l'actualitat, aquestes històries han d'ésser introduïdes manualment mitjançant el formulari de creació d'històries. Pot esdevenir un procés molt tediós i que provoca una pèrdua del voltant de 4 hores.

Els requisits són doncs:

- 2.a) Poder llegir un fitxer en format "xlsx" de Microsoft Excel.
- 2.b) Assignació dels camps del fitxer a cada variable de la taula d'històries.
- 2.c) Lectura de l'associació amb el fitxer.

⁴ Grid: graella de files i columnes.

- 2.d) Lectura recursiva de les dades a emmagatzemar a la base de dades des del fitxer.
- 2.e) Pujar el fitxer al servidor karat per a la correcta lectura.
- 2.f) Un cop el proces ha finalitzat, eliminar el fitxer usat del servidor.
- 2.g) El procés ha de ser quasi invisible a l'usuari. Únicament ha d'intervenir en la selecció del fitxer i els camps a associar.

Objectiu 3: Duplicació de Tasques.

Les tasques són la divisió de les històries en parts més petites i indivisibles que seran desenvolupades per un únic tècnic.

Sovint però, sol passar que s'obren algunes tasques a l'inici del projecte sense saber si la tasca la farà una o vàries persones, ja sigui per dificultat o necessitat de que s'afegeixi un tècnic de suport. A més a més, sovint, alguns Scrum Masters obren una única tasca de previsió i es va partint a mesura que la tasca es va desenvolupant.

Els requisits són doncs:

- 3.a) Introducció de l'opció de "Duplicar Tasca" als diferents formularis que tractin amb tasques.
- 3.b) Guardar el dia en el qual es vol partir la tasca i saber quantes hores es volen imputar.
- 3.c) Automatització al omplir el desgloss de dates amb 0 hores en les dates anteriors a la de la partició i amb les hores de la nova tasca en les posteriors.

Objectiu 4: Consulta de la imputació d'hores realitzades per tècnic/dia.

La consulta de la imputació d'hores es realitza actualment amb un fitxer Excel. Aquest Excel realitza una consulta global de les dades, per després filtrar per mostrar les hores imputades per cada tècnic i dia que formen un equip de desenvolupament i els dies compresos entre la data d'inici i final de cada sprint.

La idea bàsica és transportar aquesta funcionalitat a l'aplicació ekon SCRUM. Aquesta consulta haurà de ser ràpida, ja que únicament buscarà dades a la base de dades segons Equip de desenvolupament i número d'sprint.

- 4.a) Mòdul d'identificació d'equip de desenvolupament i número d'sprint.
- 4.b) Formatació de les dades trobades en una taula HTML.

- 4.c) Les columnes HTML hauran de seguir el patró dels dies compresos entre inici i final de cada sprint.
- 4.d) Les fileres HTML hauran de ser composades pel nom del tècnic. Únicament apareixeran tècnics que hagin estat informats.
- 4.e) És molt important que la consulta es generi de manera ràpida i neta.

Objectiu 5: Control de modificacions de les històries.

Es vol poder tenir un control de les modificacions que es realitzin en alguns camps relacionats amb les històries.

- 5.a) Detecció dels canvis en els diferents camps a controlar.
- 5.b) Guardar aquests canvis en base de dades.
- 5.c) S'ha de poder accedir de manera fàcil mitjançant un formulari en el qual es vegi un històric de canvis, ordenat per dia, i en el qual es visualitzi el tècnic que ha realitzat el canvi.
- 5.d) Si la història és esborrada de la base de dades, també han de ser esborrats els registres històrics.

Objectiu 6: Visió global Gantt dels sprints.

Es vol poder tenir una visió global dels sprints d'un equip de desenvolupament, a on es visualitzi les versions i les publicacions previstes.

- 6.a) La visió global ha de ser en forma de diagrama de Gantt.
- 6.b) La representació en forma de publicació ha de ser de tipus *milestone*, com a Micorsoft Project.
- 6.c) En el diagrama de Gantt han de ser visibles les dates d'inici i final de cada sprint.

Objectiu 7: Automatització de l'estat de les tasques/històries segons hores imputades.

En les històries i les tasques, hi ha definit un camp d'estat. Aquest camp serveix per definir el procés en el temps en el qual es troba la història o tasca. Per exemple, alguns d'aquests estats poden ser: "En curs", "Pendent", "Finalitzat", etc...

Aquest camp, però, és un camp que s'ha de canviar manualment, per tant es vol:

- 7.a) Detectar quan a una tasca ha estat finalitzada mitjançant la imputació d'hores realitzada.

7.b) Detectar quan una tasca li han sigut imputades hores.

7.c) Depenent de cada condició de imputació, actuar amb conseqüència definint l'estat pertinent.

7.d) Controlar que si una història té totes les tasques amb l'estat "Finalitzada", la història ha d'heretar aquest mateix estat.

Objectiu 8: Escriptori del Product Owner/Scrum Master.

Es vol dissenyar un escriptori (de widgets) que contingui les eines més usades pel Product Owner i/o Scrum Master, com poden ser per exemple: el gràfic de burndown, per veure l'avanç real de l'sprint, les hores de les reunions, les hores imputades per cada tècnic, etc...

8.a) Lligar tots els panells amb variables globals que identifiquin el dia en el qual ens trobem, l'equip de desenvolupament, i opcionalment l'sprint.

3.3.2. Requisits no funcionals

Els requisits no funcionals han estat establerts, amb el fi de seguir un patró comú de desenvolupament per a tots els desenvolupadors i estandaritzar l'aplicació.

- RNF1: L'aplicació haurà de poder executar-se en qualsevol sistema operatiu amb l'únic requeriment de tenir instal·lada la màquina virtual de Java.
- RNF2: Haurà d'estar desenvolupada amb tecnologia Karat⁵ per a la reducció de temps de desenvolupament i costos de programació.
- RNF3: Haurà de tenir disseny Walnut⁶ de Unit4.
- RNF4: L'aplicació s'haurà de desenvolupar amb el llenguatge de programació Java.

⁵ Karat: Plataforma tecnològica que engloba les tecnologies de Unit4, completament programable i serialitzable per a cada necessitat específica dels clients.

⁶ Walnut: Nou estil d'interfície d'usuari programable amb Java. Inclou patrons d'estandarització a nivell d'interfície.

3.3.3. Catalogació i priorització dels requisits

A continuació podrem trobar un seguit de taules que serviran per classificar els diferents requisits funcionals i no funcionals.

RF	Essencial	Condicional	Opcional
1.a	X		
1.b	X		
1.c		X	
1.d	X		
1.e	X		
1.f		X	
2.a	X		
2.b	X		
2.c	X		
2.d		X	
2.e	X		
2.f		X	
2.g	X		
3.a	X		
3.b		X	
3.c	X		
4.a		X	
4.b	X		
4.c	X		
4.d	X	X	
4.e	X		
5.a		X	
5.b	X		
5.c		X	
5.d	X		
6.a	X		
6.b	X		
6.c		X	
7.a	X		
7.b	X		
7.c	X		
7.d		X	
8.a			X

Taula 3.3. Priorització dels requisits funcionals

RNF	Essencial	Condicional	Opcional
RNF1	X		
RNF2	X		
RNF3	X		
RNF4		X	

Taula 3.4. Priorització dels requisits no funcionals

3.4. Alternatives a la solució

Com s'ha explicat amb anterioritat, ens trobem amb un projecte del qual s'han de realitzar millores. Això implica que en els inicis del projecte ja s'havien tingut en compte les diferents alternatives que hi havien.

Un cop van ser totes estudiades l'empresa Unit4 va decidir començar el projecte de desenvolupament i implementació que s'ajustes a les necessitats de l'empresa.

3.5. Conclusió

Com a conclusió, moltes de les funcionalitats a afegir són de cara a l'usuari, i per tant aquestes millores donaran un aspecte molt útil a l'usuari. Això no implica que les millores siguin necessàriament complicades, ni necessàriament fàcils de desenvolupar.

En definitiva, a la finalització del projecte, el software ha d'esdevenir una eina potent, per poder arribar a substituir el sistema de fulls d'Excel connectats a la base de dades de ekon SCRUM que són usats en l'actualitat al departament d'I+D de Unit4.

Tot projecte de desenvolupament té uns beneficis i uns inconvenients. A mode de conclusió de l'estudi de viabilitat, podem extreure els següents beneficis i inconvenients:

Beneficis:

- Millora en la productivitat de l'empresa
- Millora en l'experiència d'usuari.
- Millora en l'adquisició de gestió de dades.
- Millora en la consulta general de la informació.
- Reducció del temps en processos d'organització (reunions, plannings, etc...)

Inconvenients:

- Pot existir recel per part d'alguns usuaris.
- Pot existir un període d'adaptació dels nous usuaris.

- És possible, doncs, que sigui necessària una formació prèvia en SCRUM per poder desenvolupar amb aquesta metodologia.

Per la sèrie de beneficis i inconvenients que s'han trobat es pot concloure que els beneficis que aporta són majors que els inconvenients i per tant, el projecte **ÉS VIABLE**.

4. Pla de projecte

4.1. Introducció

En aquest capítol es realitzarà un estudi sobre el pla de projecte de les millores en l'eina ekon SCRUM en termes de realització, desenvolupament i execució. En aquest capítol es pot trobar la descripció, el WBS, els recursos del projecte, el calendari de projecte, el pressupost i també una conclusió en base als punts anteriors.

En l'elaboració del projecte, s'ha utilitzat una metodologia en espiral, ja que cada requisit analitzat i desenvolupat ha estat testejat posteriorment.

4.2. WBS

4.2.1. Fases i activitats del projecte

En la següent taula podem veure quines són les fases i activitats a nivell més general del projecte.

Fases	Descripció
Iniciació	Fase d'iniciació. Fase corresponent a la definició del projecte, assignació i matriculació
Planificació	Inclou l'estudi de viabilitat i el pla de projecte
Anàlisi	Anàlisi de requisits funcionals i no funcionals. Arquitectura del sistema.
Disseny	Inclou el disseny de la capa de dades, de control i d'interfície. Disseny dels tests.
Desenvolupament	Fase de desenvolupament de l'aplicació
Test i proves	Fase de prova del sistema. Inclou tests unitaris i d'integració.
Implantació	L'aplicació s'instal·la en el seu entorn real. Inclou la formació d'usuaris.
Generació de documents	Fase de documentació del projecte. Inclou manuals i memòria del projecte.
Tancament del projecte	Fase de tancament. El director del projecte signa l'acceptació i tancament del projecte
Defensa del projecte	Defensa del projecte davant la comissió.

Taula 4.1. Fases generals del projecte

4.2.2. Milestones

En la següent taula podem veure els punts de control del projecte. Aquests punts de control són fites importants, i per tant han de ser assolides amb la màxima puntualitat possible.

Nom	Descripció	Data
Iniciació	Matriculació	15/10/2010
Estudi de viabilitat	Aprovació	13/12/2010
Pla de projecte	Aprovació	13/12/2010
Anàlisi	Aprovació	24/12/2010
Disseny	Aprovació	07/01/2011
Tancament	Acceptació	15/05/2011
Defensa	Avaluació	06/07/2011

Taula 4.2. Milestones o punts claus del projecte

4.3. Recursos del projecte

Tot projecte té definits uns recursos humans i materials, que formen part activa del desenvolupament del projecte.

En la següent taula podrem veure quina és la valoració econòmica respecte cada recurs.

Recursos humans	Valoració
Cap de projecte	30,00 €/hora
Analista	20,00 €/hora
Programador	8,00 €/hora
Tècnic de proves	6,00 €/hora
Documentalista	6,00 €/hora

Taula 4.3. Valoració dels diferents recursos del projecte

Els recursos materials van a càrrec de l'empresa Unit4, així com el cost de les llicències.

4.4. Calendari del projecte

4.4.1. Dependències

Les fases del projecte es desenvolupen fent servir un model en espiral.

Això implica que el projecte serà dividit en els diferents requisits que hi ha, i seran analitzats i desenvolupats i testejats un a un.

És a dir, en el moment en el qual una funcionalitat hagi estat implementada, serà inclosa en l'aplicació. Com que l'aplicació ja està sent usada, això servirà per a la fase de test.

És a dir que la metodologia en espiral seguirà el procés:

ANÀLISI -> DESENVOLUPAMENT -> TEST

4.4.2. Quadre de tasques del projecte

A continuació, el quadre de tasques del projecte. Tot projecte ha de tenir una organització i definició de les tasques en el temps. La proposta de planificació està condicionada per les dates d'inici i final. El projecte es desenvoluparà des de novembre de 2010 a juny de 2011 amb una dedicació de 20 hores setmanals.

Nº/Nom de la tasca	Durada	Recursos	Pr.
1. Milllores en l'eina ekon SCRUM	200 dies		
2. Inici del projecte: assignació i matriculació	1 dia	CP	
3. Panificació del projecte	7,5 dies	CP	
4. Estudi de viabilitat	5 dies	CP	2
5. Aprovació de l'estudi de viabilitat (punt de control)	1 dia	CP	4
6. Pla de projecte	5 dies	CP	5
7. Aprovació del pla de projecte	1 dies	CP	6
8. Anàlisi de l'aplicació ekon SCRUM	31 dies		
9. Anàlisi de requisits de les milllores	15 dies	A	7
10. Anàlisi de les bases de dades actuals existens	15 dies	A	9
11. Anàlisi de l'estat actual de l'aplicació	25 dies	A	10
12. Document d'anàlisi	2,5 dies	D	11
13. Aprovació de l'anàlisi (punt de control)	0,5 dies	CP	12
14. Disseny de l'aplicació	21 dies		
15. Disseny i fraccionament modular del projecte en requeriments.	10 dies	A[50%];P[50%]	13

16. Disseny de les proves (tests)	5 dies	A	15
17. Document de disseny	5 dies	D	16
18. Aprovació del disseny (punt de control)	1 dia	CP	17
19. Desenvolupament de l'aplicació	100 dies		
20. Preparació entorn de desenvolupament - KARAT/ECLIPSE	2 dies	P[25%]	18
21. Desenvolupament modular de les millores	99 dies		
22. Millora 1: Assistent d'sprints	31 dies	P	20
23. Proves reals en planta I+D	64 dies	TP	22
24. Millora 2: Importador d'històries des d'Excel	25 dies	P	20
25. Proves reals en planta I+D	51 dies	TP	24
26. Millora 3: Duplicació de tasques	31 dies	P	24
27. Proves reals en planta I+D	35 dies	TP	26
28. Millora 4: Consulta hores/tècnic per sprint	31 dies	P	26
29. Proves reals en planta I+D	19 dies	TP	28
30. Millora 5. Control de modificacions en històries	6 dies	P	28
31. Proves reals en planta I+D	15 dies	TP	30
32. Millora 6. Visió global en Vista Gantt dels sprints	10 dies	P	30
33. Proves reals en planta I+D	4 dies	TP	32
34. Ajuda en l'adaptació de les noves bases de dades (PK removed)	20 dies	P	32
35. Disseny de dos escriptoris (opcional)	20 dies	TP	34
36. Implantació			
37. Generació del JAR final	1 dia	A[50%];P[50%]	34
38. Proves reals en planta I+D	3 dies	A[25%]; CP[25%]; P[50%]	37
39. Generació de documents	5 dies	A[50%];D	37
40. Tancament del projecte	2 dies	CP	39

Taula 4.4. Panificació prevista

4.4.3. Calendari temporal

A continuació, el quadre temporal en forma de Diagrama de Gantt. Aquí es pot veure amb una major definició, les dates i els recursos usats.

Aquest calendari temporal és el primer calendari que es fa realitzar, en el apartat 8.3 de desviacions es podrà trobar el calendari final amb les noves funcionalitats.

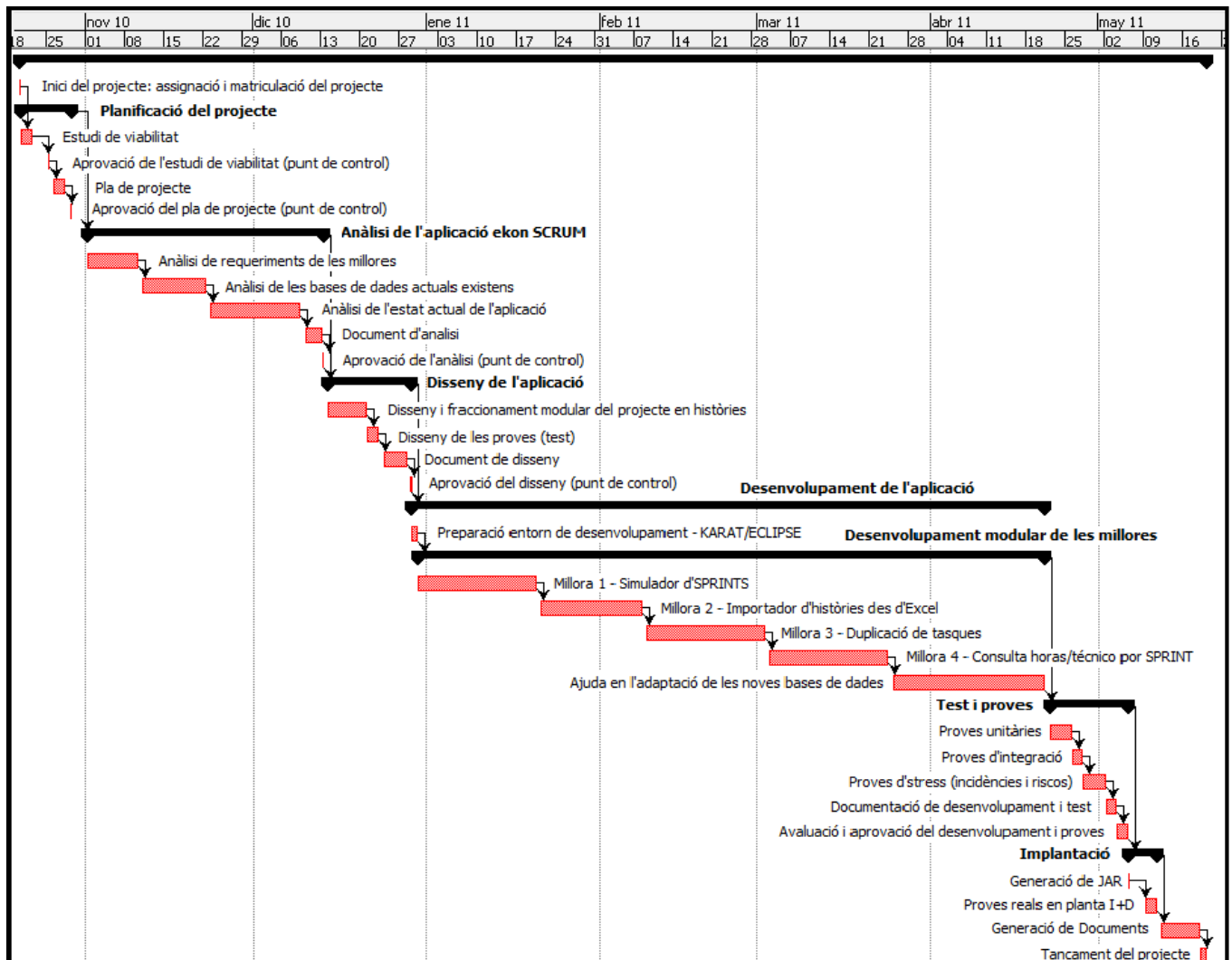


Figura 4.1. Diagrama de Gantt del projecte

4.5. Pressupost

El cost dels projectes és un paràmetre del qual no ens podem oblidar, ja que en una part molt gran, definirà la viabilitat, i el pla de projecte de manera positiva o negativa. En els següents punts veurem quines són les estimacions a nivell de cost.

4.5.1. Estimació cost de personal

L'estimació del cost del personal ha sortit de la suma de costos per cada activitat/persona de la planificació del projecte. El total és el següent:

Recursos humans	Estimació cost/personal
Cap de projecte	1550 €
Analista	5125 €
Programador	7476 €
Tècnic de proves	700 €
Documentalista	750 €
TOTAL	15601,00 €

Taula 4.5. Resum del cost del personal

El cost real que representa per a l'empresa és de 2500 €, ja que el projecte està realitzat en convenide pràctiques del qual se n'extreu que el cost total real del projecte apareix de la suma dels diners del conveni que percebrà l'alumne més el preu de la computadora destinada també per l'alumne.

4.5.2. Estimació cost dels recursos

Absolutament tos els recursos usats en el projecte, han estat de llicència lliure o de llicència interna de l'empresa.

Recursos	Llicència	Cost
Abiword	Pública general GNU	0,00 €
Eclipse IDE for Java Developers	Pública general GNU	0,00 €
Karat Studio	Llicència interna	0,00 €
Notepad++	Pública general GNU	0,00 €
OpenProj	Pública general GNU	0,00 €
SQL Developer	OTN License	0,00 €
TOTAL		0,00 €

Taula 4.6. Resum del cost dels recursos

4.5.3. Estimació cost de les activitats

Nº/Nom de la tasca	Durada	Recursos	Preu
1. Milllores en l'eina ekon SCRUM	200 dies		15601,00 €
2. Inici del projecte: assignació i matriculació	1 dia	CP	100 €
3. Panificació del projecte	7,5 dies	CP	
4. Estudi de viabilitat	5 dies	CP	500 €
5. Aprovació de l'estudi de viabilitat (punt de control)	1 dia	CP	100 €
6. Pla de projecte	5 dies	CP	500 €
7. Aprovació del pla de projecte	1 dies	CP	100 €
8. Anàlisi de l'aplicació ekon SCRUM	31 dies		
9. Anàlisi de requisits de les milllores	15 dies	A	1200 €
10. Anàlisi de les bases de dades actuals existens	15 dies	A	1200 €
11. Anàlisi de l'estat actual de l'aplicació	25 dies	A	2000 €
12. Document d'anàlisi	2,5 dies	D	100 €
13. Aprovació de l'anàlisi (punt de control)	0,5 dies	CP	50 €
14. Disseny de l'aplicació	21 dies		
15. Disseny i fraccionament modular del projecte en requeriments.	10 dies	A[50%];P[50%]	1000 €
16. Disseny de les proves (tests)	5 dies	A	500 €
17. Document de disseny	5 dies	D	500 €
18. Aprovació del disseny (punt de control)	1 dia	CP	100 €
19. Desenvolupament de l'aplicació	100 dies		
20. Preparació entorn de desenvolupament - KARAT/ECLIPSE	2 dies	P[25%]	300 €
21. Desenvolupament modular de les milllores	99 dies		
22. Millora 1: Assistent d'sprints	31 dies	P	1000 €
23. Proves reals en planta I+D	64 dies	TP	100 €
24. Millora 2: Importador d'històries des d'Excel	25 dies	P	1000 €
25. Proves reals en planta I+D	51 dies	TP	100 €
26. Millora 3: Duplicació de tasques	31 dies	P	1000 €
27. Proves reals en planta I+D	35 dies	TP	100 €
28. Millora 4: Consulta hores/tècnic per sprint	31 dies	P	1000 €
29. Proves reals en planta I+D	19 dies	TP	100 €
30. Millora 5. Control de modificacions en històries	6 dies	P	1000 €
31. Proves reals en planta I+D	15 dies	TP	100 €
32. Millora 6. Visió global en Vista Gantt dels sprints	10 dies	P	1000 €

33. Proves reals en planta I+D	4 dies	TP	100 €
34. Ajuda en l'adaptació de les noves bases de dades (PK removed)	20 dies	P	100 €
35. Disseny de dos escriptoris (opcional)	20 dies	TP	100 €
36. Implantació			
37. Generació del JAR final	1 dia	A[50%];P[50%]	100 €
38. Proves reals en planta I+D	3 dies	A[25%]; CP[25%]; P[50%]	100 €
39. Generació de documents	5 dies	A[50%];D[50%]	300 €
40. Tancament del projecte	2 dies	CP	50 €

Taula 4.7. Resum del cost de les activitats

4.5.4. Resum i anàlisi cost

En definitiva, el cost total del projecte és únicament, el cost de les activitats, ja que la resta de recursos tenen llicències GNU, o bé són internes de Unit4.

4.6. Conclusió

Aquest projecte té un cost força elevat i inicialment l'empresa no recuperarà en termes monetaris directes, la inversió realitzada.

El cost total del projecte és de 15601 €.

Això no vol dir pas, que no sigui un projecte útil, ja que com s'ha comentat en l'anàlisi de viabilitat, aquest projecte farà que l'organització dels diferents departaments de desenvolupament de l'empresa engranin i siguin autogestionats d'una manera molt més òptima en termes de temps, rendiment, i celeritat.

L'aplicació ekon SCRUM serà posteriorment venuda a altres empreses del grup *Unit4 International*, si es comprova que és una eina potent i segura.

A més a més, està planificada la venda a tercers, en el cas que es compleixi el bon funcionament internacional de l'eina. És en la venda i en l'estalvi de temps en el control del desenvolupament dels equips on es podrà recuperar la inversió realitzada en tot el procés de desenvolupament del projecte.

5. Anàlisi i disseny

5.1. Introducció

En aquest capítol s'explicaran les bases de l'anàlisi i el seu posterior disseny: quines són les funcionalitats, l'arquitectura de l'aplicació, la interfície de l'usuari, tot explicant com es realitzaran les funcionalitats i el perquè de les mateixes.

5.2. Arquitectura de l'aplicació

L'aplicació ekon SCRUM està basada en la tecnologia Karat.

L'eina karat facilita el desenvolupament de les aplicacions de l'empresa Unit4 que es desenvolupa amb OTTools.

OTTools és una eina de desenvolupament, que permet crear les bases de dades, les consultes a aquestes bases de dades, els objectes de negoci i els formularis.

La base de Karat/OTTools està programada amb el llenguatge de programació Java. Per tant, qualsevol aplicació desenvolupada amb aquesta tecnologia podrà ser executada sobre qualsevol plataforma, ja sigui Microsoft Windows, GNU/Linux, o bé MacOS. Pel que fa a les bases de dades, Karat té una capa d'abstracció que és capaç de connectar de manera transparent a l'usuari amb qualsevol de les bases de dades suportades. Per tant l'usuari té la llibertat d'escollir segons les seves preferències qualsevol de les bases de dades, com ara, MySQL, Oracle, Intersystems Cache o SQL Server.

El disseny de software amb l'eina OTTools és molt senzill i repetitiu. Es basa en:

Creació de la taula a la base de dades -> Consulta base i consulta -> Objecte de negoci -> Creació de formularis

D'una taula i una consulta base se'n poden crear n consultes, n objectes de negoci i n formularis.

La potència de l'eina OTTools radica en que aquests processos de creació són completament visuals. Ara bé, si volem tenir una personalització extra en forma de comportament de l'aplicació, es necessita escriure codi en Java associat. Aquest codi pot estar associat a l'objecte de negoci o bé al formulari.

Com a exemple, es podria citar la creació d'un botó en un formulari el qual ha de ser programat en Java per a que tingui un comportament específic i determinat.

Com a conclusió cal destacar l'alt nivell d'aquesta plataforma de desenvolupament, que deixa de banda els aspectes tècnics per a que el programador s'hagi de centrar únicament en el disseny.

5.3. Interfície d'usuari

La interfície d'usuari segueix els estàndards *Walnut*. Aquests estàndards estan definits en una sèrie de plantilles que han de ser usades obligatòriament a l'hora de fer servir la dissenyadora de formularis quan es crea un nou formulari.

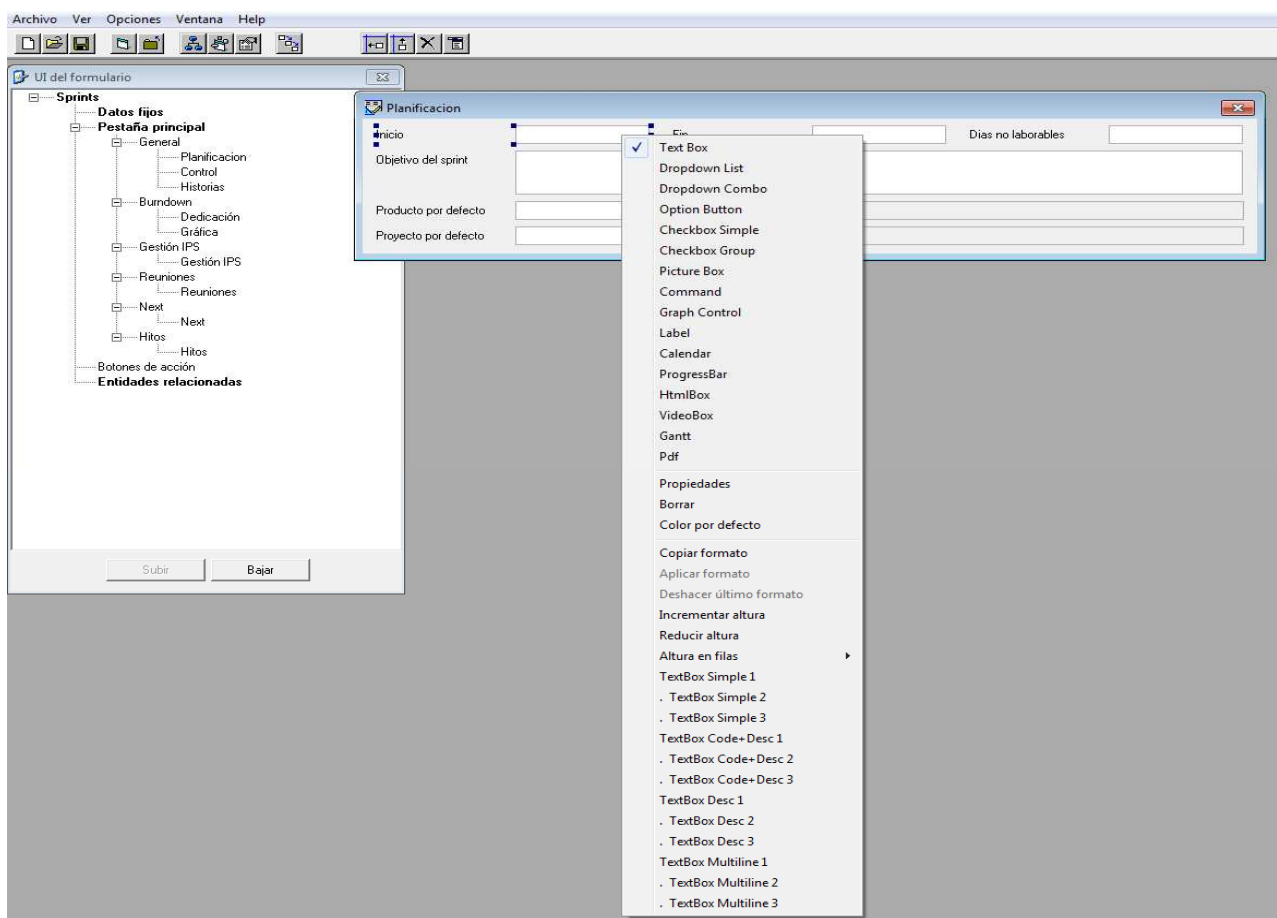


Figura 5.1. Dissenyadora de formularis

En la figura 5.1 es pot veure la definidora de formularis, i es veu com fent click secundari sobre el camp "Inicio" apareix un menú de selecció amb totes les opcions d'un camp de text en format de plantilla Walnut.

A continuació, en la figura 5.2, es pot comprovar de manera visual quin és l'estil d'interfície estàndard de les aplicacions Karat.

Producto	* Historia	* Título	Proyecto	Puntos	Prioridad	Puntos añadidos	Puntos
SCRUM	37	Reuniones Sprint 2	SCRUM	0			
TEST	441	Horas previstas	EST				
TEST	1055	Historia 1055	EST	1	1		

Figura 5.2. Vista d'un formulari Karat. Concretament el de l'assistent d'sprints.

Es pot apreciar com el formulari genera una mena de "frame" en el qual tots els camps queden alineats en desplaçament horitzontal respecte el marc esquerra, i els camps numèrics són alineats a la dreta i els camps de text a la esquerra, així com en els camps de data, karat fa aparèixer el calendari en forma d'icona, sense que el programador s'hagi de preocupar.

També està estipulat el fet que els camps de consulta del formulari que seran informats per l'usuari quedin posicionats a la part de dalt, anomenat *header*. (En aquest cas, "*Equipo*" i "*Sprint*").

Per tant, tornem a recalcar la potència mencionada en l'apartat 5.2, en la creació a molt alt nivell de qualsevol aspecte de l'aplicació, en aquest cas, els formularis.

5.4. Funcionalitats de les milllores

Es va acordar a l'inici del projecte una sèrie de milllores que farien de ekon SCRUM una aplicació molt més usable i automatitzada.

A continuació es pot trobar l'anàlisi de les funcionalitats.

- Objectiu 1: Assistent d'Sprints.

Els equips del departament I+D de Unit4 que ja treballaven amb metodologia scrum, van idear un Excel que agafava les dades de la base de dades de l'aplicació ekon SCRUM. Aquest fitxer d'Excel servia per fer una simulació de les històries que podien entrar en un sprint. D'aquesta manera es podia planificar els diversos sprints d'un projecte abans de començar el desenvolupament.

L'assistent d'sprints és la principal millora a realitzar ja que es preveu un ús intensiu per part dels Scrum Master.

L'assistent d'sprints implica l'ús de vàries taules de la base de dades, com són: la taula d'sprints, la d'històries, dedicació, next, burndown, i fites.

De la taula d'sprints es necessiten totes les dades, ja que l'assistent d'sprints les ha de visualitzar totes per poder representar un control en forma de dates, objectius de l'sprint, punts disponibles, etc.

La taula d'històries és necessària per poder relacionar les històries que anem introduint, lligant el camp sprint de la taula d'històries amb l'sprint que s'estigui tractant.

La taula dedicació és necessària per poder assignar tècnics disponibles per cada dia comprès entre l'inici i final de l'sprint.

La taula next, pot contenir històries next de l'sprint, això és, històries que són candidates a ser desenvolupades en l'sprint, sempre i quan hagin estat desenvolupades totes les històries de l'sprint.

La taula burndown serveix per contenir les dades que seran usades per generar el gràfic d'avanç de l'sprint, anomenat BurnDown⁷.

La taula de fites, serveix per poder definir fites del sprint que s'està tractant. Es poden definir vàries fites, com poden ser, "Entrega a cliente", "Pasar distribución de la publicación 1.0.1 a soporte", etc...

⁷ BurnDown: gràfic que tendeix a 0 segons les hores realitzades pels tècnics.

Pel que fa al funcionament de l'assistent, s'ha intentat replicar al màxim l'ús de l'Excel, això és:

Les dades de l'sprint més importants, han d'aparèixer en capçalera, per després poder visualitzar un *grid* o graella, per tal d'anar podent inserir les històries. Aquestes històries poden ser inserides manualment, o bé amb l'eina de *look-up*⁸, que no és res més que una consulta de les històries disponibles que són candidates a pertànyer a l'sprint que s'està visualitzant. A mesura que aquestes històries es van introduint, de manera interactiva, es van sumant els punts de cada història, i es va comparant el màxim de punts disponibles del sprint. D'aquesta manera, el scrum master pot anar veient i simulant tots els sprints del projecte, fins que totes les històries del Product Backlog hagin estat col·locades a tots els sprints.

Per tant, gràcies a l'assistent d'sprints, s'obté una gran eina de planificació en termes de la relació SPRINT->HISTÒRIA.

En principi, l'assistent d'sprints, ha de substituir l'actual formulari d'sprints, ja que l'únic que permet és definir les dades de l'sprint, però no pas associar històries amb l'sprint creat.

Pensant doncs en aquesta substitució, s'han creat altres funcionalitats que aporten un extra de dades, com són, una graella amb històries que poden ser next, i un mòdul d'inserció de fites.

Les històries next d'un sprint, no són res més que històries que poden desenvolupar-se en un sprint, sempre i quan totes les històries del mateix, hagin estat desenvolupades.

El mòdul d'inserció de fites serveix per poder definir fites, si s'escau, d'un sprint. Ja que, com hem comentat anteriorment, els sprints serveixen per poder presentar alguna cosa "tangible", ja sigui al Product Owner o al client, és interessant que els equips de desenvolupament pugin registrar quins sprints tenen aquest tipus de fita. Únicament es necessita un títol de fita i una data màxima de realització. (Aquestes fites seran visualitzades en forma de *milestone* en la Visió global dels sprints en Digrama de Gantt). En la figura 5.3 es pot veure quin és l'aspecte final del formulari d'sprints.

⁸ Look-up: opció de recerca fent click dreta al ratolí que obre una finestra amb els resultats d'una consulta que permet inserir un registre dels mostrats.

UNIT4 ekon UI - SCRUM_2000 - Ignacio Perez

Inicio Edici6n Ver Herramientas Acciones Ayuda

Búsqueda rápida

Principal Asistente (UNIT4) Sprints 1/1+

Equipo TEST - Pruebas SCRUM

Sprint 2

General Burndown Gesti6n IPS Reuniones Next Hitos

Planificaci6n

* Inicio 30/11/2010 * Fin 14/12/2010 Dias no laborables 6

Objetivo del sprint

* Producto por defecto TEST TEST

* Proyecto por defecto EST Estudio inicial

Control

Velocidad estimada 0,360 Técnicos disponibles 2,00 Puntos disponibles 6

Velocidad real 0,0 Actualizar

Historias

* Producto	* Historia * Titulo	Proyecto	Puntos	Prioridad	Puntos ańadidos	Puntos
SCRUM	37 Reuniones Sprint 2	SCRUM	0			
TEST	441 Horas previstas	EST	1	1,		
TEST	1055 Historia 1055	EST	1			

Total

Pte. Asignaci6n 6

Figura 5.3. Formulari de l'assistent sprints

- Objectiu 2: Importador de històries des de Microsoft Excel.

Aquesta funcionalitat va sorgir de la necessitat de poder introduir les dades de manera ràpida. S'ha de tenir en compte que l'entrada d'històries és molt lenta ja que la taula té 50 camps, i que per tant, entrar una única història pot ser ràpid ja que molts camps són opcionals, però el fet d'introduir-ne 400 pot esdevenir una feina molt lenta, que pot trigar al voltant de 4, 5 o fins i tot 6 hores.

Quan tot l'equip de desenvolupament es reuneix a l'inici del projecte en la reunió de sizing, es van valorant totes les històries en punts. Sol passar que en aquestes reunions el scrum master ja porta un fitxer Excel amb totes les històries enumerades en el moment de l'anàlisi del projecte.

En acabar aquestes reunions, es té un fitxer Excel, amb les dades de les històries.

Gràcies a l'eina importadora d'Excel, es pot relacionar cada columna del fitxer Excel amb un camp de dades de la taula d'històries, i així d'aquesta manera es pot inserir a l'aplicació en un temps mínim (5 o 6 segons), totes les històries.

Aquesta eina, permet els modes d'inserció i actualització. Al fer inserció, es demanarà obligatòriament informar les columnes del fitxer que corresponguin amb les claus primàries més uns quants camps que són obligatoris, com el títol i el propietari de la història. Al realitzar una actualització de les històries, únicament es demanaran les columnes que tinguin camps de clau primària, i aquella sèrie de camps que l'usuari vulgui actualitzar.

Figura 5.4. Formulari de la importació de històries

- Objectiu 3: Duplicació de Tasques.

Es desitja poder duplicar tasques sense alterar els totals previstos ni les imputacions ja realitzades. Cal recordar que les tasques són individuals i són particions d'una determinada tasca. Cada tasca és auto-assignada per un sol tècnic. L'aplicació contempla el fet de que una tasca té certes hores pendents que van sent restades a mesura que es van imputant hores realitzades. Ara bé, sovint les tasques són obertes a l'inici de cada sprint sense saber si les farà una persona o vàries. Aquestes tasques són anomenades internament com a tasques de previsió, en les quals són informades les hores que es creu que es trigarà a desenvolupar la història. És a dir, per una història, només es crea una tasca (de previsió). A mesura que la història es va desenvolupant, la tasca es partint en tasques més petites per poder imputar hores i tècnic.

Per tant, un cop la història ha estat finalitzada, la tasca d'inici ha estat partida en n tasques necessàries per cada tècnic i/o requisit de desenvolupament de la història. També pot ser útil aquesta partició/duplicació, per tasques en les quals intervé un segon tècnic de suport i aquest necessita imputar hores (pel principi de: una tasca = un tècnic).

El procediment que segueix el formulari de partició de tasques és senzill.

Es selecciona la tasca origen que es vol duplicar, es sol·licita la data de partició (la qual ha d'estar dintre de l'sprint en el qual pertany la història), les hores que haurà de tenir la nova tasca a crear.

Un cop es tenen aquestes dades, el procés obrirà una nova tasca amb les hores realitzades a 0 i el desgloss d'hores a realitzar abans de la data de partició serà de 0 hores i a partir de la data de partició serà omplert amb les hores informades anteriorment de la nova tasca. Un cop fet això, el procés de duplicació restarà les hores de la tasca nova en la tasca origen des de la data de partició fins al final.

Aquest formulari, serà accessible des d'un punt de menú, des de la definidora de tasques i des del seguiment d'hores de les tasques mitjançant un botó d'acció des dels dos formularis mencionats.

Es pot veure un exemple a la figura 5.5.

UNIT4 ekon UI - SCRUM_2000 - Ignacio Perez

Inicio Edición Ver Herramientas Acciones Ayuda

Búsqueda rápida

Principal

Gestión SCRUM

- Product Backlog
- Historias
- Propuestas
- Sprints
- Tareas
 - Duplicar
 - Imprimir tarjetas
 - Visualización paneles
- Horas
- Reuniones
- Incidencias
- Enlace ADMIN
- Informes
- Administración
- Configuración
- Consulta global
- Seleccionar empresa activa

karat starter kit

karat starter kit edición Lite

Escritorios

Favoritos

Datos

* Producto AGORA WALNUT

* Historia 1 Walnut en Ágora

* Tarea 1 diseño MBL

Tarea origen

Descripción diseño MBL

Tipo ANA

Técnico

Previsión 4

Número IPS

Fecha inicio sprint 26/03/2011

Fecha final sprint 30/04/2011

Horas pendientes 0

Duplicación Tarea

Nueva tarea a crear 3

Descripción diseño MBL

Técnico

Número IPS

Fecha de partición 29

Horas de la nueva tarea

Aceptar Cancelar

Duplicar

(UNIT4) Importación de datos (UNIT4) Duplicación de Tareas

Figura 5.5. Formulari de la duplicació de tasques

Objectiu 4: Consulta de la imputació d'hores realitzades per tècnic/dia.

Aquest requisit esdevé d'una utilitat creada mitjançant full d'Excel recollint dades de la base de dades de l'aplicació, en la qual es generava unes graelles filtrades en relació a l'equip i sprint seleccionat en les quals es podia veure una relació de les hores realitzades per cada tècnic/dia.

D'aquesta manera es pot visualitzar amb una consulta com van les hores del sprint, sense haver de fer servir el fitxer Excel, ni tampoc anant als formularis d'imputació d'hores i realitzar manualment la suma d'hores de cada tècnic.

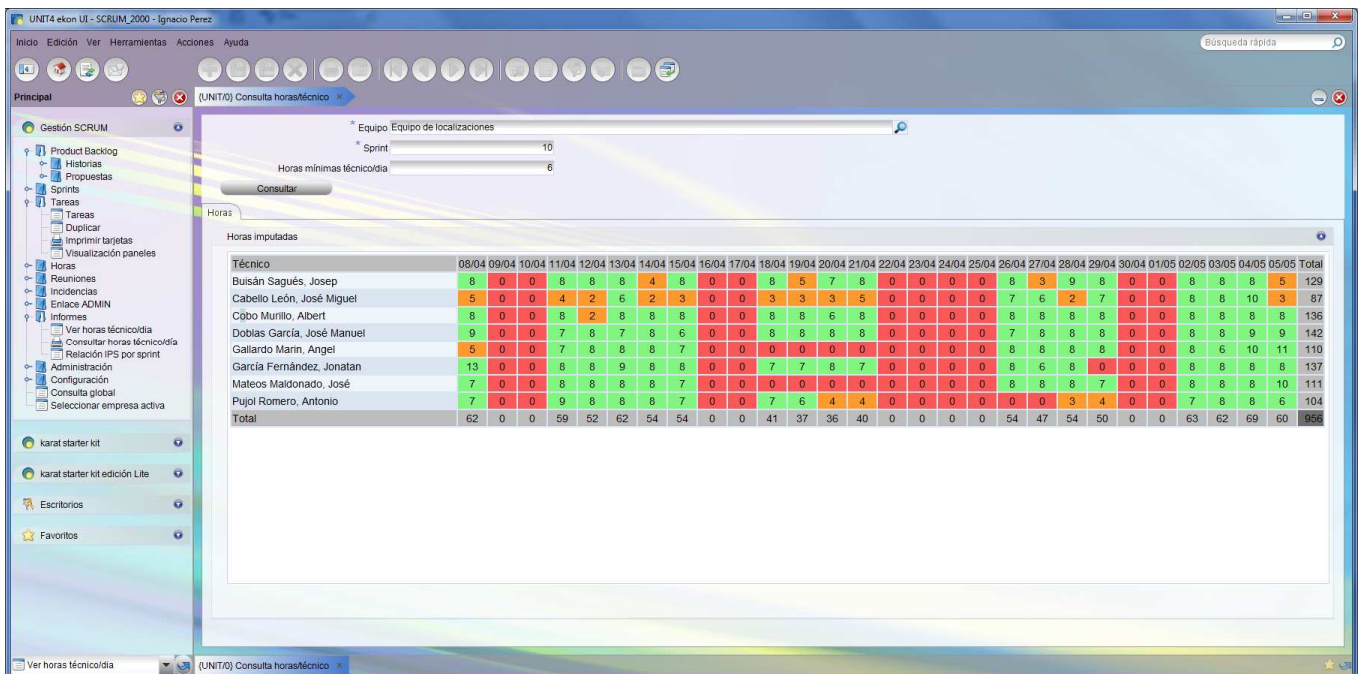


Figura 5.6. Formulari de la consulta d'hores per tècnic i dia

Objectiu 5: Control de modificacions de les històries.

Un dels problemes que van veure alguns scrum masters, és el poc control que es té sobre els canvis produïts en alguns camps crítics de les històries.

Per això es va decidir estudiar quins camps se n'hauria de portar un control històric, i es va resoldre que els camps a controlar serien "Deseo", que serveix per especificar d'una manera més concreta què es vol obtenir al desenvolupar la història. El segon camp a controlar es diu "Para que", que serveix per controlar i especificar el perquè de la història i quin possible benefici pot tenir un cop s'hagi desenvolupat la tasca. I el tercer i últim camp a controlar és el camp "Comentario", que serveix per acabar de polir detalls no especificats en altres camps.

Per perfilar aquesta funcionalitat, es fa servir un procés que va vigilant els canvis de qualsevol d'aquests tres camps, i es guarden en una taula de la base de dades en la qual s'especifica la història, el producte, la data i hora del canvi, els camps que han variat, i l'usuari que ha fet el canvi.

D'aquesta manera es pot evitar qualsevol canvi no controlat, i que s'acabi desenvolupant una història que no tindrà res a veure amb el propòsit inicial.

Cal remarcar que el control històric comença des del primer registre que s'ha entrat a la història.

Figura 5.7. Formulari de creació de històries. Es pot visualitzar el botó de "Visualizar historial"

Figura 5.8. Formulari del històric dels camps a guardar

Objectiu 6: Visió global Gantt dels sprints.

Per poder veure d'una manera global el calendari dels sprints amb les seves respectives fites, es demana poder visualitzar un simple diagrama de Gantt.

Mitjançant l'eina karat, es pot representar en un formulari mitjançant codi en XML. Això permet anar llegint les dates d'inici i final dels sprints de l'equip seleccionat, així com les dates de les fites.

Amb totes aquestes dades es poden anar pintant les barres representatives temporals per a cada sprint, i les fites en forma del típic *milestone* tant usat en programes de planificació com són Microsoft Project o OpenProj.

Per tant, per poder pintar el diagrama de Gantt únicament necessitem la taula d'sprints i la taula de fites de la base de dades.

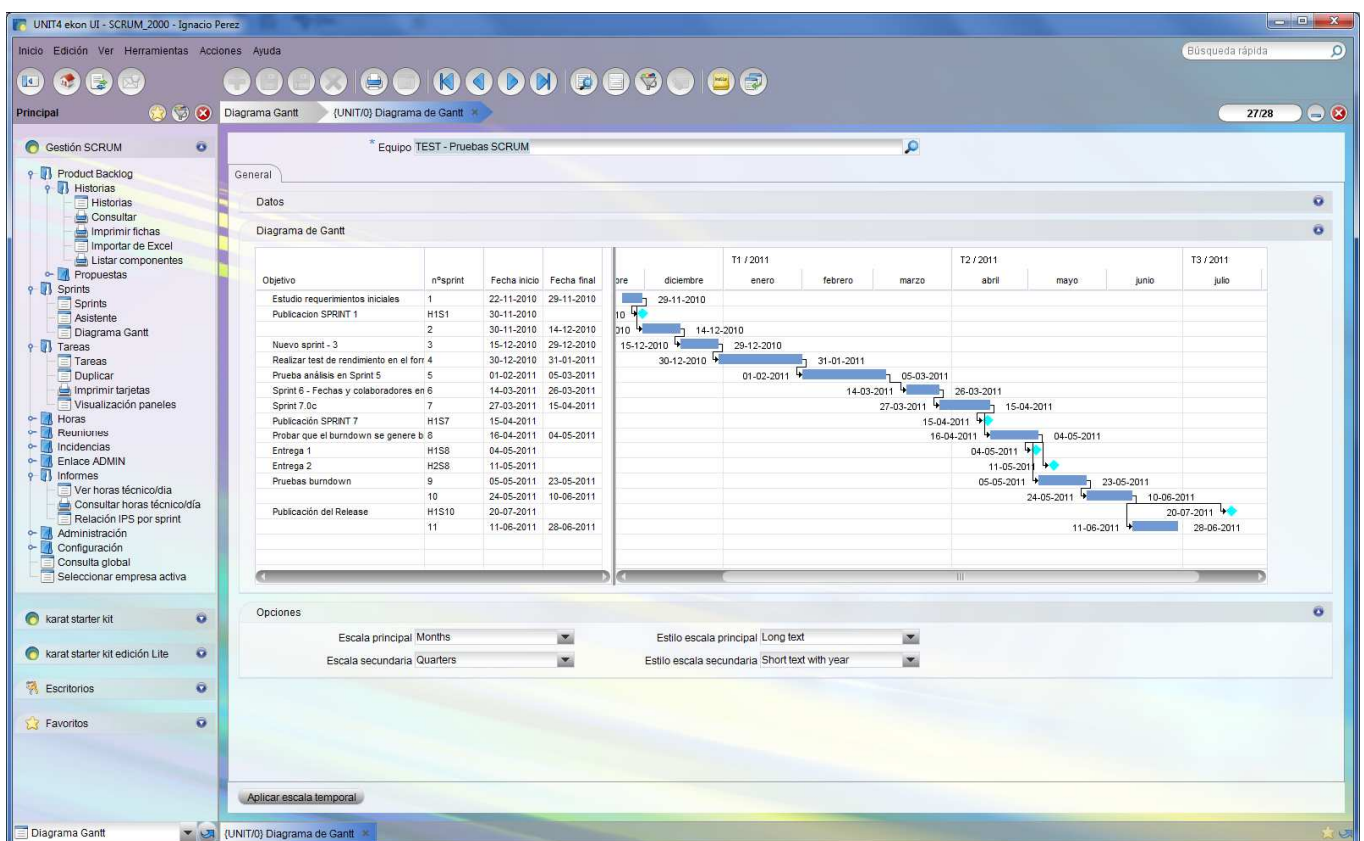


Figura 5.9. Formulari de la visió global dels sprints

Objectiu 7: Automatització de l'estat de les tasques/histories segons hores imputades.

Aquest requisit va sorgir d'una recomanació d'un becari que va treballar l'any passat en l'aplicació.

Tal i com s'ha comentat anteriorment una història té n tasques que la conformen.

Tant les tasques com les històries tenen un camp d'estat. Aquest camp d'estat indica en quin procés del desenvolupament es troba en termes d'hores realitzades.

Així per exemple, quan una història té totes les tasques sense hores imputades per part dels tècnics, l'estat de la història està fixada en "Pendent", de la mateixa manera que totes les tasques que la conformen també tenen l'estat "Pendent".

El problema era que quan un tècnic imputava hores treballades en una tasca, aquesta no variava el seu estat de manera automàtica, sino que el propi tècnic havia de variar manualment el seu estat, i havia de variar també l'estat de la història de la qual penja la tasca. L'algorisme ha implementar ha de ser capaç de detectar les hores imputades per qualsevol tècnic a les tasques, així com les hores pendents de realitzar per finalitzar la tasca. Dependent del resultat d'aquesta detecció, l'estat de la tasca serà "Pendent", "En curs", o bé "Finalitzada".

En cada canvi d'hores imputades detectat per l'algorisme, també s'ha de vigilar l'estat de totes les tasques com a conjunt. Així d'aquesta manera, si de totes les tasques que hi ha en una història, una de les tasques té l'estat "En curs" encara que la resta de tasques tinguin estat "Pendent", l'estat de la història també haurà de ser "En curs".

Un cop totes les tasques han estat finalitzades, aquestes obtenen l'estat "Finalitzat", així com l'estat de la història mare que també ha de quedar fixat a "Finalitzat".

Tarea Descripción	Tipo	Colaborador	Previsión	Estado	Número IPS
1 Reuniones varias	REU		16	Acabado	

Fecha	Horas realizadas	Horas pendientes	Comentario
15/12/2010	0	16	
16/12/2010	1	15	
17/12/2010	0	15	
18/12/2010	0	15	
19/12/2010	0	15	
20/12/2010	6	5	
21/12/2010	7	0	
22/12/2010	0	0	
23/12/2010	0	0	
24/12/2010	0	0	
25/12/2010	0	0	
26/12/2010	0	0	
27/12/2010	0	0	
28/12/2010	0	0	
29/12/2010	0	0	

Figura 5.10. Formulari de la imputació d'hores a tasques. L'estat es varia automàticament.

Pel que fa a l'escriptori, s'ha desestimat la seva implementació, perquè no suposaria una millora essencial, no gaire funcional, ja que la idea del escriptori és la de poder varis formularis oberts en una mateixa pantalla, i en l'anàlisi, no s'ha trobat un ús recursiu de varis formularis alhora paral·lelament.

5.5. Diagrama de la base de dades

La base dades de l'aplicació conté moltes taules, però les millors que s'han realitzat únicament comporten l'ús d'algunes taules i un parell de nova creació essencials per les millors a realitzat.

Les taules usades són:

- cs_products: taula amb les dades dels productes desenvolupats de l'empresa.
- cs_projects: taula amb els projectes que es desenvolupen segons l'empresa.
- cs_team: taula amb els equips de desenvolupament de l'empresa.
- cs_sprint: taula amb totes les dades necessàries per informar un sprint.

- cs_stories: conté les històries dels diferents equips de desenvolupament, lligades amb equip i sprint.
- cs_workitem: taula que conté les tasques d'una història en particular.
- cs_workflow: taula que conté un registre amb cada dia del sprint en el qual està la història inclosa.
- cs_stories_hist: taula que guarda el registre històric dels canvis en alguns camps de les històries.
- cs_sprint_hitos: aquesta taula serveix per guardar les fites de cada sprint.
- cs_next: bàsicament es una còpia de la taula de històries, però aquestes històries seran incloses en el next (històries que seran desenvolupades si queda temps a l'sprint) dels sprints.

Les tres últimes taules han estat creades en el projecte actual. Les anteriors han estat retocades respecte l'any passat per un canvi en les claus primàries a nivell global de l'aplicació.

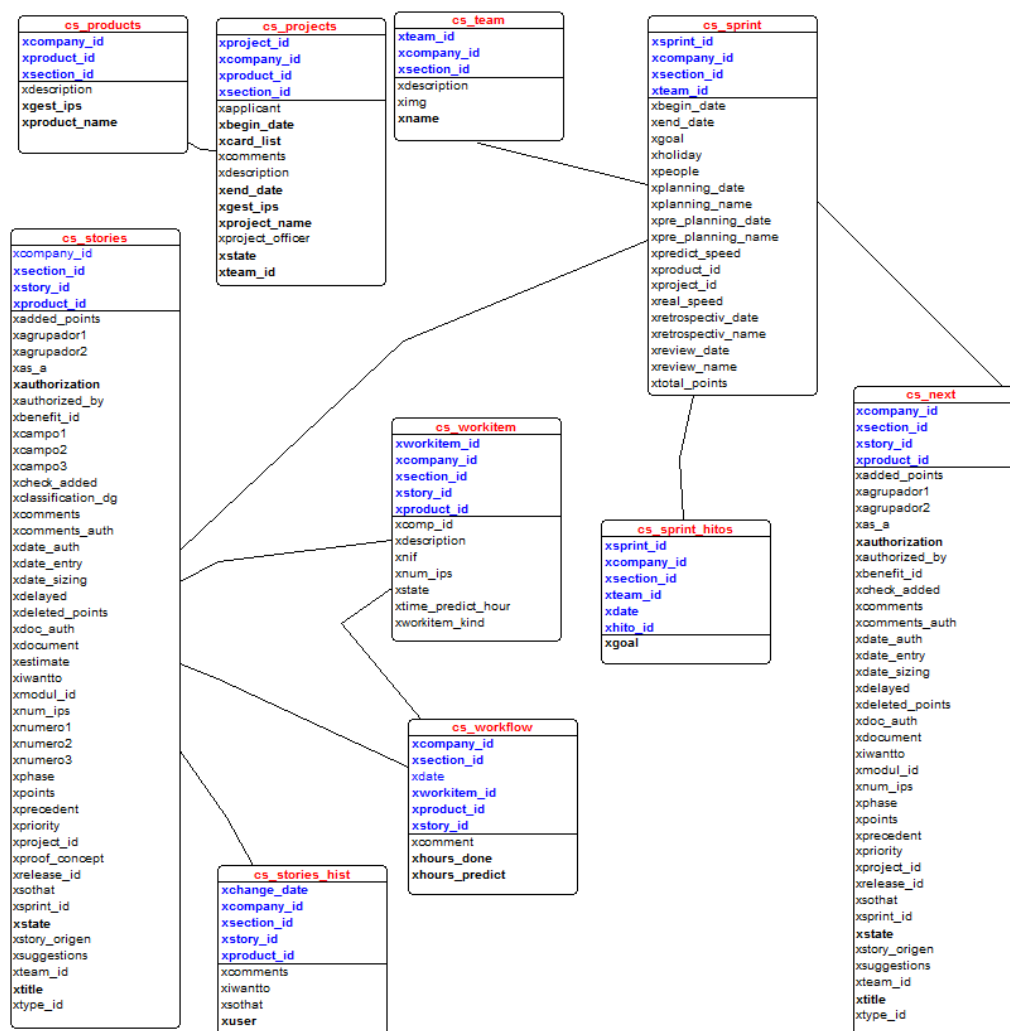


Figura 5.11. Diagrama de les taules usades

6. Implementació

6.1. Introducció

En aquest capítol s'explicarà tot allò relacionat amb la implementació. En els següents punts doncs, es podrà trobar la informació relacionada amb els entorns de desenvolupament i d'execució així com algunes normatives de l'eina karat per realitzar un desenvolupament correcte en Java.

6.2. Entorn de desenvolupament

L'entorn de desenvolupament està definit en dos bandes. Per una tenim OTTools i per l'altre Eclipse.

Eclipse és un IDE, *Integrated Development Environment* per les seves sigles en anglès, que es pot traduir per entorn de desenvolupament integrat.

OTTools, com s'ha comentat anteriorment (cap 5.2), és una eina integrada que serveix per definir consultes a bases de dades, objectes de negoci associats a aquestes consultes i formularis associats a aquests objectes de negoci.

La combinació d'aquestes dues eines permet la creació i personalització del software executat en plataforma karat.

Per tant, arribats a aquest punt ens podem fer dues preguntes. Es pot programar software karat únicament amb OTTools? I a més, es pot programar software karat únicament amb Eclipse? La primera de les preguntes té resposta afirmativa, ja que OTTools permet a cop de click, dissenyar consultes, objectes de negoci, i formularis que són la base fonamental d'un software karat. El problema d'aquestes aplicacions realitzades amb OTTools esdevé en la poca usabilitat que poden arribar a mostrar, ja que no tenen un nivell de personalització elevat, i en definitiva són molt estàtiques. En canvi, la segona de les preguntes té una resposta negativa. No es pot desenvolupar una eina karat únicament amb codi Java desenvolupat amb Eclipse.

Dins la plataforma karat, està previst que el llenguatge Java serveixi com a una eina de personalització.

És a dir, per dotar de funcionalitats avançades no es pot aconseguir d'una altra manera que amb Eclipse i Java.

Per tant, es podria dir que OTTools està pensat per al disseny en forma de taules, objectes de negoci, formularis i Eclipse/Java per a la implementació d'aquestes.

Com a exemple, es pot citar una de les millors realitzades, l'Importador de històries Excel. Amb OTTools, es generen els objectes de negoci que contenen les variables de la taula de històries i el formulari amb tots els botons i camps necessaris per al correcte funcionament. Un cop realitzat això, és necessari crear una classe Java que personalitzi el funcionament de la funcionalitat esmentada.

6.3. Entorn d'execució

L'entorn d'execució està basat en la típica arquitectura Client-Servidor. Tots els clients, tant remots com locals, es connecten a un servidor que és el que té el codi de personalització Java i l'estructura de les dades de OTTools, com poden ser les taules, les consultes, els objectes de negoci i els formularis. Aquest servidor alhora també té les taules i les dades per al correcte funcionament de l'aplicació.

Sol ser costum, que el servidor al qual es connecten els clients sigui únicament l'encarregat de servir d'intermediari i és el que gestiona les connexions amb un altre equip que té realment la una base de dades amb la definició dels objectes d'OTTool, anomenat servidor d'aplicacions, i un altre amb les dades, anomenat servidor de base de dades.

Aquests dos servidors, el de base de dades i el d'aplicacions, poden estar en un mateix equip o en dos equips diferents.

Els requisits dels diferents equips que interactuen en l'execució són els següents:

- Client karat:

Els clients karat amb Linux i Windows amb processadors Intel 32 i 64 necessiten una CPU multi-core 2.4 GHz o superior i 2 GB de memòria RAM o superior. Si el client s'executa en Mac es necessita una CPU G4 1.0 GHz o superior o CPU Intel Core Duo i 2 GB de memòria RAM o superior.

- Servidor:

Pel servidor de base de dades es recomana una CPU XEON multi-core 3.0 GHz o superior, 4 GB de memòria RAM o superior, Xarxa de 1 GB entre servidor d'aplicacions i servidor de base de dades, i una infraestructura adequada per a que es puguin connectar els clients.

Pel servidor d'aplicacions, sempre i quan només es dediqui a aquest servei, es recomana exactament el mateix que el servidor de base de dades.

OTTools crea dues bases de dades necessàries per al funcionament de les aplicacions. Aquestes bases de dades es diuen SDIC i DATA. La SDIC és la base de dades que conté les dades de la creació i generació de les consultes, objectes de negoci i formularis. Aquesta base de dades no està pensada per a que es manipuli per l'usuari, sinó que simplement l'eina OTTools llegeix o escriu a la taula segons el desenvolupador realitzi canvis mitjançant OTTools.

La base de dades DATA serveix per poder llegir i/o guardar les dades d'usuari de l'aplicació. Pot ser modificada pel desenvolupador per poder corregir possibles errades en fase de desenvolupament sense afectar el funcionament de l'aplicació ja que simplement són dades independents del funcionament de l'aplicació.

Per tant, com s'ha comentat anteriorment amb els requisits de les màquines del client, la plataforma karat client pot ser executada en tots els sistemes operatius.

6.4. Desenvolupament en Java

Gràcies al codi Java, podem personalitzar el comportament de les aplicacions karat. La base de la personalització de karat està basada en la programació dels events que la pròpia aplicació genera.

Internament karat va generant events, i es feina del desenvolupador identificar els mateixos per poder-los programar amb l'objectiu de canviar-ne el comportament. La gestió d'aquests events segueix uns passos concrets.

En IDE Eclipse, es crea un nou projecte karat associat al software que es vol desenvolupar. Per cada formulari o objecte de negoci a personalitzar s'ha de crear una classe Java associada com es pot veure en la

figura 6.1. Per assignar una classe java als objectes de OTTools és necessari indicar a l'objecte de negoci i/o al formulari la ruta exacte mitjançant la propietat dels mateixos, tal i com es pot veure a la figura 6.2.

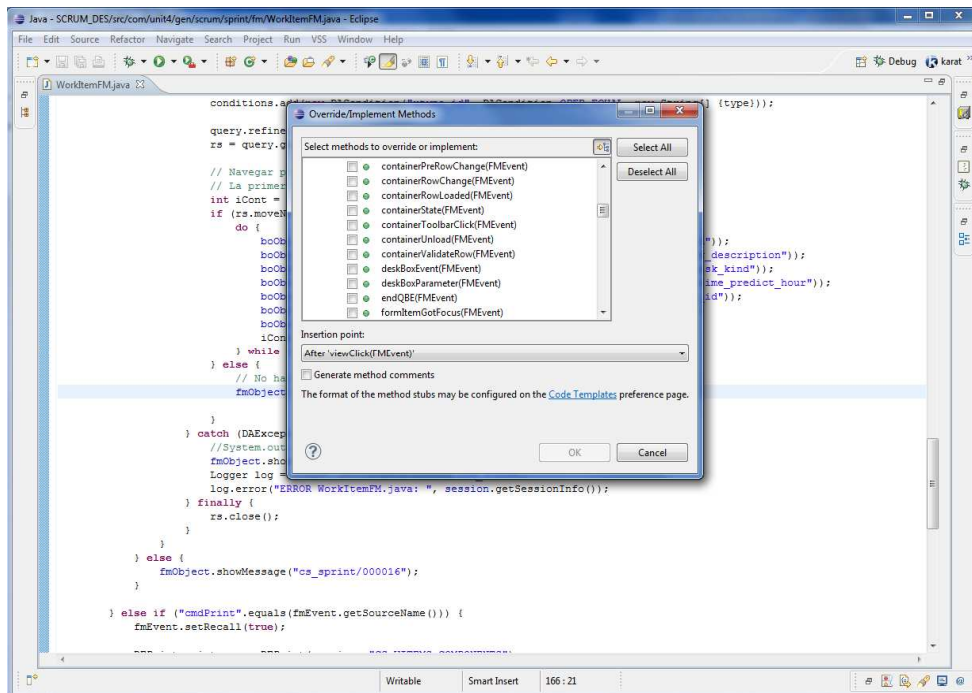


Figura 6.1. Mètodes a desenvolupar segons els events que genera karat.

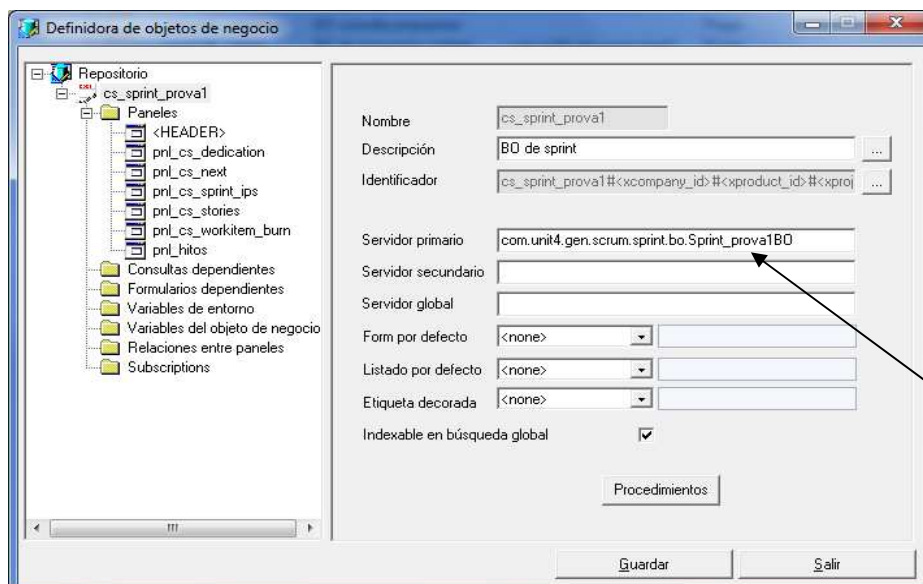


Figura 6.2. Propietats de l'Objecte de negoci associant en el camp "Servidor primario" la ruta del codi Java

7. Codificació i proves

7.1. Introducció

En aquest capítol s'explicarà l'estil de codificació usat en el desenvolupament i les proves que s'han realitzat. Pel que fa a les proves, s'han realitzat proves unitàries i per poder comprovar que tot queda completament integrat s'han realitzat proves d'integració.

7.2. Estil de codificació

Pel que fa al tema de l'estil de codificació, no hi ha una norma escrita al respecte, tot i que els projectes solen fer servir l'estil de codificació anomenat *Kernighan&Ritchie/Kernel* que està basat en el següent exemple:

```
if(condicio) {
    sentencia0;
    sentencia1;
}
```

Personalment per al desenvolupament realitzat al projecte s'ha fet servir el tipus de codificació *BSD/Allman* del qual podem veure un exemple a continuació:

```
if(condicio)
{
    sentencia0;
    sentencia1;
}
```

La diferència fonamental és la posició de la clau de l'obertura del mètode i/o condició de flux. En el primer estil es pot veure com la clau està situada al costat de la sentència i en el segon estil la clau està situada just a sota de la condició.

Personalment, a costa d'una línia extra s'obté més claredat de l'inici i finalització dels fluxos i dels mètodes.

A Unit4 hi ha una normativa en estil de codificació en relació al nom de les variables i mètodes que poden haver en un codi. Aquest estil implica que tots els noms de les variables han de començar per minúscula i si la variable ha de tenir més d'una paraula les posteriors a la primera començaran per majúscula. Per exemple, “String nomFitxerExcel”.

Exactament el mateix passa amb el nom dels mètodes. Per exemple:

“public void formLoad(FMEvent fmEvent)”

7.3. Proves unitàries

Com que l'aplicació s'ha fet servir durant tot el projecte al departament I+D, les noves funcionalitats han estat provades en una sèrie de dades creades per tal de no modificar dades d'altres equips de desenvolupament.

Aquestes dades estan basades en un equip de desenvolupament anomenat "Equip de TEST", i les històries que es creaven estaven assignades al projecte "EST" que són lletres de la paraula TEST.

Cada vegada que una funcionalitat ha estat implementada, s'han realitzat proves unitàries per comprovar el seu correcte funcionament. Això és, fer servir les dades del prova, per comprovar que les dades han estat guardades, actualitzades o esborrades correctament. Alhora, també s'han realitzat proves amb valors límit per comprovar que això no provocava cap error.

A continuació exposem una llista de les proves realitzades a les funcionalitats implementades:

- Proves a l'assistent de sprints:

Les proves realitzades en l'assistent han implicat bàsicament el canvi massiu de històries d'un sprint a qualsevol altre, que és la funcionalitat primordial de l'assistent. Per tant, en la prova s'han creat 2 sprints i 10 històries degudament informades, però sense sprint definit. Un cop realitzat això, totes les històries han estat introduïdes mitjançant l'assistent al sprint 1.

Cada vegada que una història és afegida a un sprint, es genera una graella amb les dates des de l'inici del sprint fins al final, per tal de poder imputar hores a les tasques que conformen cada història. Acte seguit d'incloure les històries al sprint es comprova, que aquestes graelles han estat ben generades

amb les dates correctes. Sempre que una història sigui moguda d'un sprint a un altre, és necessari que el procés esborri les dates anteriors de la graella i en generi les noves corresponents al sprint de la història.

També és necessari comprovar que quan una història té tasques amb hores imputades, aquella història queda fixada al sprint i impedeixi el moviment de la mateixa entre els diferents sprints.

Les proves van concloure que hi havia errades en aquest últim cas, ja que l'assistent no impedia la introducció d'una història amb hores imputades en un altre sprint. Aquest error ha estat corregit.

- Proves al importador de històries des de Microsoft Excel.

El importador és la funcionalitat més crítica de l'aplicació ja que implica l'entrada de dades des d'una font externa, en aquest cas un fitxer d'Excel. El propi procés que es segueix en la importació ja comporta la comprovació de la integritat de les dades, però sempre es poden generar errades. En aquesta prova s'ha fet servir un fitxer d'Excel amb valors límit per veure quin és el comportament de l'aplicació. Aquest fitxer conté valors de tipus text en camps únicament numèrics. En total el fitxer conté al voltant de 50 històries. És a dir, unes 50 línies ja que cada línia és una història.

La prova va concloure que si es trobava un error per exemple en una història que estava en la línia 35, l'aplicació entrava les dades correctament fins aquella línia que contenia les dades errònies i ja no n'importava cap més. Això era difícil de controlar, ja que l'usuari no sabia quines eren les històries que havien entrat correctament i quina era la línia que provocava l'error.

Això es va solucionar controlant la transacció de la inserció de les dades. Així quan la inserció o actualització comença, les dades es van guardant en memòria fins arribar al final del fitxer, i si les dades són totes correctes la transacció es completa i s'insereix a la base de dades. Pel contrari, si hi ha algun error en el fitxer Excel, cap de les històries és introduïda a la base de dades i s'informa quin és l'error que ha provocat la no inserció de les dades.

- Proves en la duplicació de tasques.

Les proves realitzades en la duplicació de tasques es refereixen al funcionament en sí. Per tant, la prova s'ha realitzat des d'una tasca origen, comprovant que la data de duplicació de la nova tasca està dintre de les dates del sprint d'on penja la història que conté la tasca origen. També es comprova que la tasca origen tingui les hores correctes, corresponent a la resta de les hores amb les que es crea la nova tasca amb les hores que queden per realitzar a la tasca des del dia de la partició fins a la data final.

Les proves en la duplicació no han trobat cap error a corregir.

- Prova de la consulta de la imputació d'hores realitzades per tècnic/dia.

En aquesta funcionalitat, al tractar-se d'una consulta, les proves a realitzar són únicament de rendiment. El problema que es va trobar en les primeres proves és que el formulari trigava molt a generar la graella amb les hores de cada tècnic. Aquest mal funcionament en termes de rendiment esdevenia pel plantejament de l'algorisme, ja que per cada dia i tècnic s'executava una query restringida per aquests dos registres esmentats.

Es va decidir crear una query que retornés tots els valors necessaris i després formatejar la graella de manera adequada per a cada registre.

En la consulta que més trigava a generar la graella es trigava 25 segons i amb el nou algorisme implementat el temps es reduïa a 4 segons.

- Proves de l'automatització de l'estat de les tasques/històries segons hores imputades.

Les proves en el canvi d'estat s'han realitzat en una única història i tres tasques que pengen de la mateixa. La història i les tasques han estat inicialitzades amb l'estat "Pendent".

Els diferents canvis d'estat poden ser deguts, tenint en compte 1 història amb 3 tasques, a:

- Si les tasques no tenen hores imputades l'estat de les mateixes i de la història és "Pendent"
- Si una o dues tasques tenen hores imputades, aquestes varia o varien l'estat a "En curs" de la mateixa manera que l'estat de la història.

- Si una o dues tasques tenen les hores completades, aquestes varia o varien l'estat a "Acabat" i la història segueix amb l'estat "En curs".
- Si les tres tasques tenen hores imputades, l'estat és "En curs" de la mateixa manera que l'estat de la història. I si totes les tasques tenen les hores imputades necessàries per a finalitzar la història, l'estat de les tres passen a "Acabat" i la història també.

Les proves realitzades no han trobat errors en els canvis d'estat.

- Proves en el control de modificacions de les històries.

Les proves en el control de modificacions fan referència al funcionament. L'algorisme implementat detecta canvis en tres camps de la taula històries i en guarda un històric. Les proves, doncs, han estat basades en la modificació dels tres camps, "Deseo", "Para que" i "Comentarios", per comprovar que realment es quedin guardats els nous canvis.

Un camp de la clau primària de la taula del històric de les històries és la data del canvi, i l'altre és l'usuari. Es va trobar un problema quan un mateix usuari realitzava un canvi en els tres camps alhora i el procés detectava que havia de guardar a la taula els registres ja que es generava un error de clau primària ja que l'usuari era el mateix i la data/hora també.

Aquest problema trobat va ser solucionat amb la inclusió al codi d'un retard manual realitzat amb un *thread.sleep(1000)* que provoca una diferència d'un segon entre registre i registre i d'aquesta manera la clau primària de la data/hora ja es variava per tal de que no estigués duplicada.

- Proves en la visió global Gantt dels sprints.

En les proves de la visualització dels sprints en forma de Gantt, no s'han trobat errors, ja que és només una visualització estàtica dels sprints (data inici i data final), amb precedència entre sprints i les fites col·locades en forma de *milestone* entre mig dels sprints segons la data de cada fita.

7.4. Test en departament I+D

Tal i com s'ha comentat en el punt anterior (7.2), l'aplicació ja s'estava fent servir des de que l'any passat els 4 becaris van finalitzar els seus respectius projectes i l'aplicació ekon SCRUM es va finalitzar amb èxit.

Això ha implicat que totes les millores desenvolupades en l'actual projecte han estat incloses en l'aplicació per a que tots els usuaris les poguessin provar.

Per tant, des de la primera funcionalitat, s'han anat corregint errors trobats del propi ús de l'aplicació, que poden arribar a ser els errors de funcionament no trobats en les proves unitàries.

L'ús de l'aplicació al departament per part dels usuaris finals ja ha implicat la realització de la prova d'integració ja que cada usuari necessita de tots i cada un dels formularis per fer servir l'aplicació, i si algun usuari trobava un error o suggeriment, aquesta era corregida o implementada.

8. Valoració del projecte final de carrera

8.1. Objectius aconseguits i no aconseguits

Totes les millores previstes han estat realitzades amb èxit. L'únic punt que s'ha desestimat ha estat la realització dels escriptoris, ja que gràcies a les funcionalitats implementades no ha estat necessari crear un escriptori amb els formularis de l'aplicació en versió minimitzada.

Això ha estat suplert amb un bon disseny dels punts de menú que esdevé amb una organització concreta que facilita l'ús seqüencial dels formularis, segons l'ordre natural del procés scrum. Això és, introducció de històries, assignació de les històries als sprints i posteriorment creació de les tasques necessàries en cada història i a continuació totes les funcionalitats específiques per a la imputació d'hores a tasques i les consultes informatives, com pot ser la consulta d'hores imputades.

En canvi, un requisit que s'ha afegit i que ha estat també un dels motius de no realitzar l'escriptori, és el de realitzar documents d'ajuda associats als formularis de l'aplicació. Aquests documents s'obren mitjançant la tecla F1. Les hores estipulades per realitzar l'escriptori han estat destinades a realitzar aquests documents d'ajuda.

Per tant, l'assistent de sprints, el importador de històries Excel, la duplicació de tasques, la consulta de la imputació d'hores, el control de modificacions de les històries, la visió global dels sprints, l'automatització de l'estat de les històries i tasques i els documents d'ajuda, així com altres retocs que han estat necessaris en altres parts de l'aplicació per al correcte funcionament de les funcionalitats esmentades en la llista, han estat desenvolupades i provades amb èxit.

8.2. Possibles ampliacions i millores

En els dos últims mesos de desenvolupament, quan ja estaven finalitzades les noves funcionalitats i ja havien estat incloses a l'aplicació, el meu tutor Ramon Torres, va convocar un parell de reunions amb l'assistència dels scrum masters (caps d'equip) de tots els equips del departament I+D. La primera va ser el 14 d'abril i la segona i última va ser el 19 de maig.

La primera reunió va servir per obtenir una primera impressió de les noves funcionalitats i obtenir una sèrie de suggerències i errors a implementar per acabar de polir el funcionament de l'aplicació.

Quan tots els punts acordats en l'acta de la primera reunió es van implementar i es va comprovar que funcionaven correctament, es va deixar passar un temps per a que els usuaris, que ja havien estat avisats dels canvis, provessin l'aplicació. Per tant, el 19 de maig es va realitzar la segona i última reunió de scrum masters, per tornar a presentar les millores de ekon SCRUM i analitzar si encara quedaven coses pendents.

El resultat va ser positiu, i l'aplicació va quedar tancada a noves funcionalitats. Com que l'aplicació ekon SCRUM és de moment un producte intern de l'empresa, en la reunió es va acordar donar accés als scrum masters per a que ells mateixos realitzin les millores que creguin convenientes sempre i quan hi hagi unanimitat.

Un punt que no s'ha tingut en compte és el de la seguretat. Del projecte de l'any passat hi havia desenvolupat un mòdul de seguretat horitzontal, però aquest, generava molts problemes i es va decidir eliminar, per tant l'aplicació no té cap tipus de seguretat. Seria interessant afegir un mòdul de seguretat per tal de que cap usuari malintencionat esborri o visualitzi dades crítiques. Però al tractar-se d'una aplicació d'ús intern no és un requisit molt urgent.

Així que es pot concloure que l'aplicació queda tancada i completament funcional, subjecte a possibles millores que es puguin suggerir o possibles errors futurs que no hagin aparegut fins la data de tancament.

8.3. Desviacions respecte la planificació

Els objectius inicials del projecte es van anar ampliant a mesura que s'anaven implementant ja que la celeritat amb les que les millores es van dur a terme van fer possible la inclusió de nous objectius.

De fet, a la presentació inicial del mes de gener a Unit4 estava previst desenvolupar l'assistent, la importació de històries des d'Excel, la duplicació de tasques i la consulta de la imputació d'hores realitzades per tècnic/dia. Per tant, es va afegir al projecte el control de modificacions en històries i la visió global dels equips en forma de diagrama de Gantt.

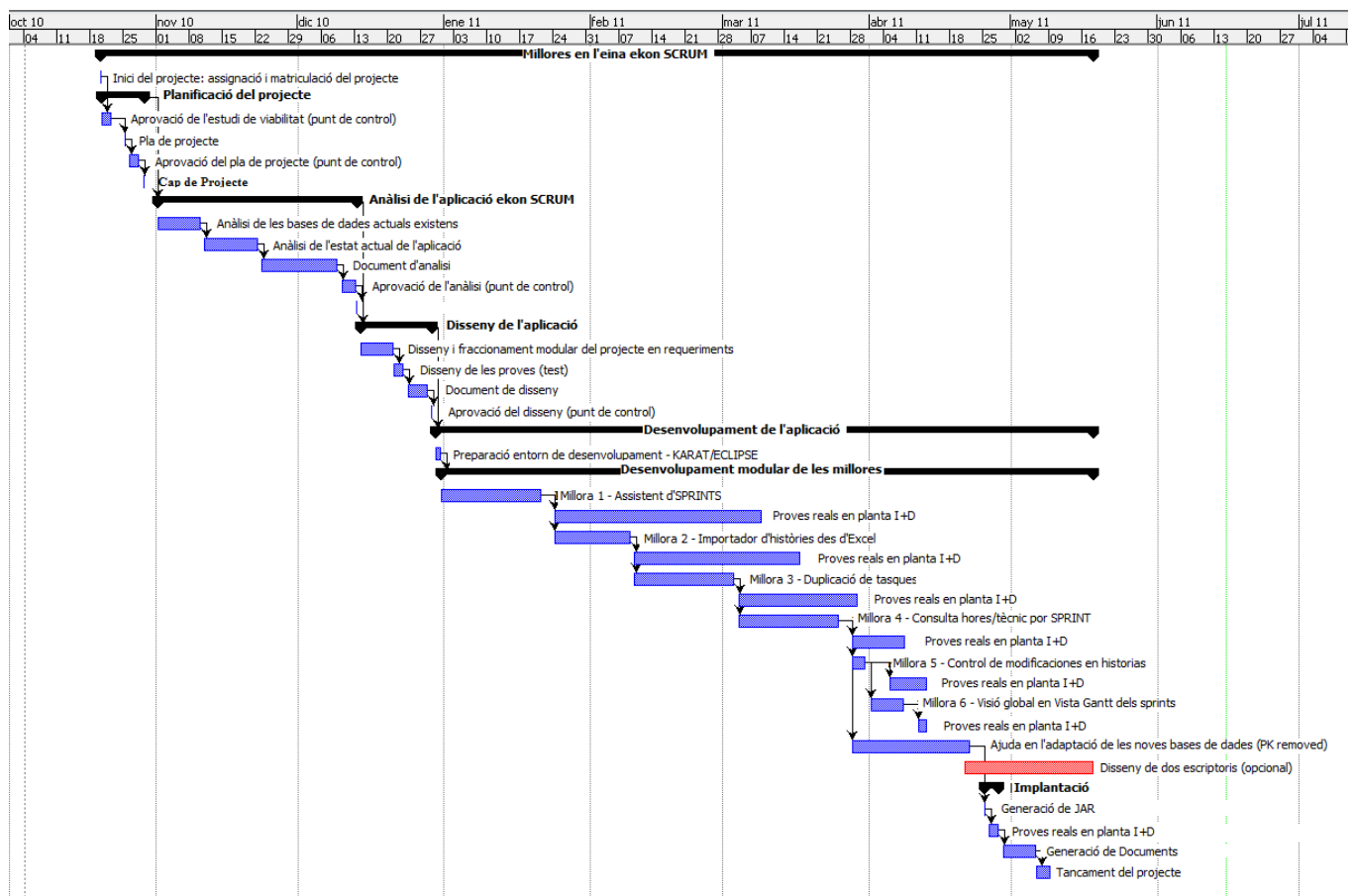


Figura 8.1. Diagrama de Gantt de la planificació final.

8.4. Valoració personal

Durant la meua estada a l'empresa he conegut en primera persona el procés de desenvolupament de software, assimilant al mateix temps la metodologia Scrum. Això m'ha aportat una visió professional que d'una altra manera no hagués tingut realitzant un projecte final de carrera a nivell personal. Cal remarcar que l'ambient de feina ha estat molt bo i gràcies al suport de la gent el projecte ha estat completat amb èxit. En definitiva recomano que altres alumnes no deixin passar una oportunitat tan bona com aquesta per poder fer el projecte final de carrera en aquesta o en qualsevol altra empresa ja que així es pot finalitzar la carrera amb una bona experiència professional en l'àmbit informàtic.

A més a més, és quasi obligatori comentar que la plataforma karat/OTTools és una eina impressionant que facilita moltíssim la feina al programador ja que la fase de disseny esdevé casi automàtica gràcies a les eines de disseny d'objectes de negoci i formularis amb patrons *walnut*, de la mateixa manera que el disseny de les taules.

També comentar que la personalització del events de karat també és molt bona, ja que es poden personalitzar quasi tots els events imaginables.

Pel que fa al projecte "Millores en l'eina ekon SCRUM" crec que ha estat un projecte molt interessant. S'ha de tenir en compte que plasmar en un software totes les necessitats de l'empresa en torn a la metodologia scrum és un procés complicat i que al finalitzar el projecte s'han pogut substituir els fitxers Excel que tiraven de la base de dades gràcies a les noves funcionalitats implementades.

Per part de l'empresa, l'assoliment dels objectius inicials i la possibilitat d'haver-los ampliat ha fet que l'eina ekon SCRUM com a projecte hagi quedat tancada, amb un bon grau de satisfacció respecte l'eina i citant paraules textuais d'en Ramon Torres: "Per part de l'alumne hi ha un bon grau d'implicació, demostrant proactivitat i fent suggeriments per la millora de las funcionalitats i en resum, amb satisfacció per part de l'empresa respecte al treball realitzat"

Bibliografia:

Llocs d'Internet:

- www.proyectosagiles.org
Pàgina web amb documentació molt variada sobre SCRUM, amb els seus principis, casos d'èxit, exemples, beneficis.
- www.wikipedia.org
Utilitzada en consultes vàries.

Documents interns d'ajuda Unit4:

- **API de karat**
Especificació API per a la plataforma karat, versió 8.0 build 4197 (API: Application Programming Interface).
- **¿karat 8.0 & eclipse? ... ¿Pero si es muy fácil**
Manual d'ajuda i iniciació al funcionament de Eclipse amb karat.
- **SCRUM en 5 minutos**
Document d'ajuda sobre el mètode scrum, explicat en un breu document amb les bases de la metodologia.
- **Introducción a Java**
Llibre de distribució a Unit4 amb les bases de Java.
- **Directrices Walnut**
Guia per a la correcta implementació del disseny de formularis en estil walnut.

Contingut del CD:

- Memòria del projecte en PDF.
- Videos de l'aplicació en funcionament.
- Exposicions i documents de reunions de UNIT4.
- Exposició inicial realitzada a Metodologia i Gestió de Projectes.
- API de karat.
- Codi de l'aplicació empaquetat en format JAR.
- Manuals de karat/Eclipse.

Firmat: Ignacio Pérez Caro

Sabadell, juny de 2011